

DAS MAGAZIN FÜR COMPUTER-FANS

For ever young: Geos

- 10 Jahre Geos - wie es anfing und wie's weitergeht
- Brandneue Geos-Kreationen der PD-und Shareware-Szene
- Erstes DFÜ-Spiel unter Geos für 25 Teilnehmer: Trade & War (auf Disk im Heft)

Assembler generalüberholt

AssBlaster 3.0:

- Neueste Version mit vielen zusätzlichen Funktionen
- Ab sofort kompatibel zu Turbo- und VisAss

Universelles Grafik-Tool

- Studio de Luxe: Entwerfen Sie blitzschnell eigene Spiele-Landschaften!

Diskette im Heft

Sofort im Bilde: Alle Assembler-Befehle des 6510-Prozessors im C 64 auf einen Blick!

PUBLIC DOMAIN

aktuelle PD-Software:

Spiele, Adventures, Strategie, Unterhaltung, Simulationen, Rollenspiele, Action, ...
Anwenderprogramme, Datenbanken, Textverarbeitung, Kalkulationen, Utilities, Sound, Grafiken, ...
Lernprogramme, GEOS-PD, Demos, 128er Software, ...
auf beidseitig bespielten Disketten **ab DM 2,-** / 5,25"-Diskette!

FD2000 3,5"-Software für FD 2000 - Laufwerke! Fordern Sie bitte die FD-Liste an - **GRATIS.**

FARBBÄNDER

Komp. Qualitäts-Druckerfarbbänder, Farbe: schwarz, lagerfähig verpackt.

Art.:	Druckertyp:	Gruppe:	Preis:
F02	Citizen 120D, Swift24		9,50
F30	Citizen Swift 24,4-farb.		23,50
F03	Comm.MPS 801	628	8,90
F04	Comm.MPS 802	629	10,50
F31	Comm.MPS 803	624	9,90
F02	Comm.MPS 1200	621	9,50
F32	Comm.MPS 1224	663	12,90
F01	MPS 1224, 4-farbig		18,90
F05	Comm.MPS 1230	673	12,50
F33	Comm.MPS 1500	674	17,90
F37	MPS 1550 C, 4-farbig		27,50
T83	Comm.1270,Tintenpatr.		29,90
F08	Epson FX, RX 80	635	8,90
F10	Epson LX 80, 90	639	7,90
F35	Epson LQ 100	659	11,50
F21	Seik. SP180,1600	678	10,90
F22	Star LC10, LC20	692	7,90
F24	Star LC 24-10	691	9,50
F40	Star LC 24-10, 4-farbig		18,90
F41	Star LC 200		9,90
F42	Star LC 200, 4-farbig		24,50
F25	Star NL10, NB 24-10		11,50

MODULE

The Final Chess Card	DA	39,-
Nordic Power-Cartridge	DA	59,-
Action-Cartridge MK6	DA	**99,-**
Utilitydiskette zum MK6	DA	29,-
Erweiterungsdisk. MK6	DA	19,-
Pagefox (das DTP-Modul) (mit 100 KByte Speicher)	DA	228,-

SONSTIGES

Diskettenlocher 5,25"	7,50
Leerdisketten 5,25" 2DD	5,00
Reinigungsdisk. 5,25"-Laufw.	9,90
100er-Diskettenbox 5,25"	14,90
500 Endlos-Etiketten, 89x35	8,50
Druckerkabel (Userp.-Centr.)	27,90
Joystick Scorpion Junior	14,95
Joystick Scorpion + TP137	19,95
Joypad (Tecno Plus) TP511	19,00
Original-Scanntronik-Maus (voll 1351- / GEOS-kompatibel)	59,00

ACHTUNG: Geos-Neuauflage!

Folgende Titel sind ab Ende August endlich wieder lieferbar:

Geos 2.5 C-64	DV	89,-
GeoFile C-64	DV	59,-
GeoCalc C-64	DV	59,-
GeoPublish C-64	DV	59,-
GeoChart C-64	DV	49,-
Deskpack / Geodex	DV	49,-
Geos 2.0 C-128	DV	119,-
GeoFile C-128	DV	79,-
GeoCalc C-128	DV	79,-

Katalog 1995

bitte kostenlos und unverbindlich anfordern!

SOFTWAREPAKETE

Sparpaket (50 Top-Progr.)	10,-
Riesenpack (100 Programme)	19,-
Actionpack (33 Actionspiele)	19,-
Adventurepack (33 Adv.)	19,-
Strategiepack (33 x Strategie)	19,-
Unterhaltungspack (33 x Unt.)	19,-

LERNPROGRAMME

C-64: Der Einstieg	EI528	10,-
Amateurfunkpack	AF565	10,-
Astronomiepack	AP515	10,-
Mathepack	DM556	10,-
Schreibmaschinenkurs	SC557	10,-
Engl.-Wörterbuch (10000 Vok.)		10,-
Englischpack (26 Lektionen)		19,-
Vokabeltrainer Französisch		10,-
Vokabeltrainer Italienisch		10,-
Vokabeltrainer Latein		10,-
Vokabeltrainer Russisch		10,-

ANWENDUNGEN

Haushaltsbuchführung	HB560	10,-
Finanzpack (19 Prog.)	FP513	10,-
Tabellenkalkulation	TK568	19,-
Multi-Lager (Lager-/Kundenverw.+Fakturierung)	DV	29,-
Multi-Datei-System (universelles Datenverwaltungssys.)	DV	29,-
99 Anwenderprogramme für		22,-
Kombitext (Textverarb.)	KB516	19,-

C-64 ORIGINALE C-128

AKTUELLE SPIELESOFTWARE

Diskette

5,25"-Disk. für Floppy 1541/71:

Adventure Collection (Soul Crystal, Spirit of Adv., Crime Time)	DV	**49,-**
Afterburner (Flugsimulator)	EV	29,-
Alternative World Games	DA	10,-
BERANIA - Der Kampf (Rollenspiel, 64er 1/95: 8 von 10 - **NEU!**)	DV	**49,-**
Big Box 2 (30 Top-Spiele)	DA	29,-
Bundesliga Manager	DV	**39,-**
California Games	EV	29,-
Cartoon Collection (Dizzy,Spike in Transylvania,CJ's Elephant, ...)	DA	29,-
Clik Clak (Geschick)	DA	29,-
Colossus Chess 4	DA	10,-
Deutsches Afrika Korps	DV	49,-
D.A.K. Szenario Disk.	DV	19,-
Die Prüfung (Rollenspiel)	DV	29,-
Energie-Manager (nur C-64)	DV	5,-
Erben des Throns (Strategie)	DV	39,-
Eskimo Games (Sport)	DA	29,-
Flightsimulator 2 (Flugsim.)	DV	79,-
Football Manager 3	DA	29,-
Gremlins (Geschicklichkeit)	EV	19,-
Gunship (Hubschraubersim.)	DA	**39,-**
Heavenbound (Jump'n'Run)	DA	**19,-**
Hook (Peter Pan / Plattform)	DA	29,-
High Five (Thalamus) (Creatures 1, Snare, Retrograde, ...)	DA	49,-
Ice Hockey (Simulation)	DA	10,-
Invest (Handelssimulation)	DV	19,-
Krieg um die Krone 1 (Str.)	DV	19,-
Krieg um die Krone 2 (Str.)	DV	29,-

GAME-KOFFER: NEU!
Zak Mc Kracken, Oil Imperium und European Soccer DV **29,-**

Lethal Weapon (Film-Action)	DA	29,-
Lemmings NEU!	DA	**39,-**
Locomotion (Züge steuern)	DV	39,-
Lords (ähnlich Populous)	DA	25,-
Lords of Doom (Horrorsim.)	DV	29,-
Mayhem in Monsterland	DA	**35,-**
Mixed Collection (Crime Time,Lords of Doom,Rolling Ronny,...)	DV	**49,-**
Motley Tetris NEU!	DA	**10,-**
Nick Faldo Golf (Golfsim.)	DA	39,-
No.2 Collection (Winzer, Black Gold und Super Soccer)	DV	**49,-**
Ormus Saga 1 (Rollenspiel)	DA	19,-
Ormus Saga 2 (Rollenspiel)	DA	19,-
Ormus Saga 3 (Rollenspiel)	DA	29,-
Outrun (Autorennen)	EV	29,-
Riddles and Stones NEU!	DV	**20,-**
Rings of Medusa (Rollensp.)	DV	39,-

5,25"-Disk. für Floppy 1541/71:

Robin Hood (Adventure)	DV	10,-
Rolling Ronny (Jump'n'Run)	DV	29,-
Scenario Theatre of War (Strategie-Simulation / 1.Weltkrieg)	DV	49,-
Schwert & Magie 1-8 (Adv.)	DV	49,-
Sleepwalker (Jump'n'Run)	DA	29,-
Spy vs Spy (die zwei Spione)	DA	10,-
Stories of Beryland 1	DA	19,-
Stories of Beryland 2	DA	19,-
Streetfighter 2	DA	39,-
Super Soccer (Starbyte)	DV	19,-
Sword of Honour NEU!	DA	**29,-**
The Manager Collection (Invest,Transworld,Black Gold,Super Soccer)	DV	**49,-**
Tom & Jerry 2 (Jump'n'Run)	DA	29,-

TURRICAN-SUPERPACK: NEU!
Turrican 1+2 **+ Joystick** DA **29,-**

Wrath of the Demon	DA	39,-
WWF Wrestling 2	DA	39,-
Zak Mc Kracken (Grafikadv.)	DV	29,-

Spielen wie im Fernsehen:

Riskant	DV	19,95
Wetten, daß ...?	DV	19,95
Dingsda	DV	19,95
Bingo	DV	19,95
Hopp oder Top	DV	19,95
Glücksrad	DV	19,95
Punkt, Punkt, Punkt	DV	19,95

ACHTUNG! 5,- DM-Angebot:

5 th Gear (Autorennen)	DA	5,00
Artura (Fantasy / Action)	DA	5,00
Blue Thunder (Helikopter)	DA	5,00
Bulldog (Weltraum-Action)	DA	5,00
Challenge Golf (Golf-Sim.)	DA	5,00
Chopper Commander	DA	5,00
Cosmic Causeway	DA	5,00
H.A.T.E. (Weltraum-Action)	DA	5,00
Highway Encounter	DA	5,00
Kellogg's Tour (Fahrrad)	EV	5,00
The Muncher (Riesen-Dino)	DA	5,00

1581-Format

3,5"-Disketten für Floppy 1581:

Black Gold (Kohleförderung)	DV	29,-
Deutsches Afrika Korps+Szen.		49,-
Krieg um die Krone 1	DV	19,-
Rebel Racer (Geschick)	DA	19,-
Rings of Medusa (Rollens.)	DV	39,-
Robin Hood (Adventure)	DV	10,-
Rolling Ronny (Jump'n'Run)	DV	29,-
Scenario Theatre of War	DV	49,-
Sparpaket (50 Programme)	DA	10,-
Transworld (Güterverkehr)	DV	19,-
99 Anwenderprogramme für		22,-

Cassetten

(DATASETTE erforderlich!)

Air Sea Supremacy (Silent Service,Gunship,F-15,Carrier Comm.)	DA	29,90
Baby of Kangaroo	DA	4,90
Chart Attack (Lotus,Supercars, ...)		14,90
G-LOC (Flugsimulation)	DA	9,90
Grand Monster Slam	DA	4,90
Kick Off 2 (Fußball)	DA	4,90
T.Mutant Hero Turtles 1	DV	4,90
T.Mutant Hero Turtles 2	DV	4,90
X-Out	DA	4,90

VERSANDKOSTEN:

bei Vorkasse (bar, V-Scheck)	5,-
per Nachnahme (incl. aller Gebühren)	10,-
Ausland (nur Vorkasse: bar / EC-Scheck)	15,-

Wir liefern auch Software+Zubehör für: AMIGA, PC und Atari.
Fordern Sie die entsprechenden Infos **GRATIS** an.
Bitte unbedingt das jeweilige System angeben!
Alle Angebote solange der Vorrat reicht.

DATA HOUSE

NEU !!! Ladengeschäft + Versand
Inh. Kai-Uwe Dittrich
Harleshäuser Str. 67 - **34130 Kassel**

Telefax: 0561 - 68405 **Telefon: 0561 - 68012**



Sämtliche Produkte erhalten Sie auch in folgenden Ladengeschäften:
Commodore-Service Genzel: Am Anger an der B 84, 99947 Kirchheilingen
Computertechnik Müller & Co.: Göttinger Str. 18, 37308 Heiligenstadt
Zur 48: Hauptstr. 54, 04416 Markkleeberg

ACHTUNG! Wir sind umgezogen!

INHALT 9/95





Viele Spitzenprogramme des C 64 kommen aus den Bereichen Public Domain/Shareware. Wir haben im aktuellen Angebot gestöbert ... **8**

Zehn Jahre Geos – 1985 fiel der Startschuß zur Entwicklung der grafischen Benutzeroberfläche. Das erste Ergebnis konnte sich sehen lassen: Geos 1.2 **10**



44 *Ausgereift und ausgefeilt – so präsentiert sich das gewaltige Software-Paket rund um den Super-Assembler "AssBlaster": die Toolstation 5.1, jetzt mit Reassembler und Konverter*

Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind

Aktuell

Grundlagen

Spieletests

Public Domain - Shareware

Geos

Tips & Tricks

Programmieren

Anwendungen

Rubriken

Wichtiger Hinweis:
Aus Gründen der Inkompatibilität zu anderer Software auf der Programmservice-Disk können wir die Dateien zu "Studio de Luxe" (S. 42) erst auf der Diskette zur 64´er 10/95 veröffentlichen.

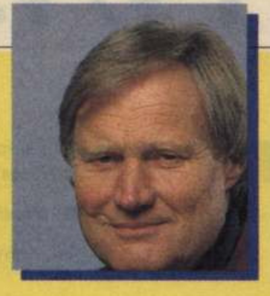
Seite 10

Seite 15

Seite 44

Seite 42





Herrliche Geos-Welt

Ein Jubiläum jagt das andere: zuerst feierte der C 64 selbst sein 10jähriges, anschließend die 64'er – Ihr Magazin –; dann reihten sich jede Menge Soft- und Hardware-Großhändler in den illustren Kreis ein (z.B. Goodsoft, Scanntronik usw.).

In diesem Jahr ist eines der besten Software-Produkte dran, das jemals für den C 64 entwickelt wurde: die komfortable Benutzeroberfläche Geos (Graphic Environment Object System).

1985 galt der Apple Macintosh als Non-Plus-Ultra in der Computer-Szene: mit dessen Benutzeroberfläche inkl. Icons, Gadgets und übersichtlichen Symbolen (Piktogramme/Icons) fühlte sich der Anwender im siebten Computer-Himmel. Das ließ eine Gruppe ausgefuchster Computer-Freaks nicht ruhen: Im August 1985 verwirklichte Brian Dougherty seine Idee von der grafischen Benutzeroberfläche für den als kleine Sensation eingestuften Commodore-Computer C 64 (zumindest, was Sound- und Grafikfähigkeiten betraf).

Rasch wurde das neuartige System vom User dankbar angenommen – schnell entwarfen die Entwickler von Berkeley Softworks deshalb professionell angehauchte Applikationen: GeoWrite, GeoCalc, GeoFile, GeoChart und GeoPublish. Damit konnte man damals schon bedeutend komfortabler arbeiten als mit vergleichbarer Software auf den PCs. GeoWrite beispielsweise wartete als Trendsetter mit Funktionen und Features auf, die man erst viel später auch in Microsofts Textverarbeitung "Winword für Windows" wieder entdecken sollte. Und GeoPublish bot bereits alle Standards, um ansehnliches Desktop-Publishing mit dem C 64 zu realsieren, die dem beliebten DTP-Programm für Macintosh-Computer verblüffend ähnelten – und das lange, bevor Pagefox auf den Markt kam.

Es war schon bewundernswert, wie die Programmierer mit dem Mini-RAM des C 64 hantierten (knapper Objekt-Code, Overlay-Verfahren usw.). Die Situation besserte sich schlagartig, als Commodore 1987 seine RAM-Erweiterungsmodule (zu damals horrenden Preisen) in Umlauf brachte. Jetzt endlich wurde die Arbeit mit Geos zum wahren Vergnügen – die Anpassung der grafischen Benutzeroberfläche an den Flash8-Mikroprozessor in jüngster Zeit ließ Geos nahezu so schnell werden wie Windwos auf dem IBM-kompatiblen AT.

Ständig wurde an der System-Software gefeilt und verbessert – findige Hardware-Entwickler und -Hersteller taten das ihre dazu, um Geos bis heute in unverminderter Beliebtheit am Leben zu erhalten. Bis Jahresende soll uns ja die brandneue Patch-Version zu Geos 3.0 ins Haus stehen – wir sind gespannt darauf wie ein Flitzebogen!

Ihr

Harald Beiler

Harald Beiler
Chefredakteur



CMD Direkt Sales senkt Festplatten-Preise

Eine kräftige Preisreduzierung auf dem Harddisk-Sektor gibt CMD Direkt Sales bekannt: Das sind die aktuellen Typen des CMD-Sortiments:

- HD 40, 660 Mark
- HD 85, 719 Mark
- HD 170, 799 Mark
- HD 340, 899 Mark
- HD 540, 999 Mark
- HD 1000, 1499 Mark

Weitere Neuheit im Verkaufsprogramm: die Expansionsport-Erweiterung EX2+1 (identisch mit EX3 - Test s. 64'er 8/95 -, allerdings besitzt die EX3 zwei vertikale Ports und einen horizontalen). Die EX2+1 kostet 79 Mark.

Bestellungen sind bei CMD oder in der MATTING-Box möglich.

CMD Direkt Sales, Postfach 58,
A-6410 Telfs/Österreich

Jugendwettbewerb via Internet

Mehrere japanische High-Tech-Unternehmen haben sich zusammengeschlossen, den "GII (Global Information Infrastructure) Junior Summit '95" auszurichten. Das Jugend-Gipfeltreffen steht unter dem Motto "Rettet die Zukunft der Jugend – rettet die Welt". Weltweit sind Jugendliche von zwölf bis 18 Jahren aufgerufen, sich zu beteiligen. Voraussetzung ist die Zugriffsmöglichkeit zum "Internet". Der Text muß unbedingt in Englisch verfaßt sein.

Verlangt wird eine maximal vierseitige Arbeit mit Vorstellungen bzw. Lösungsvorschlägen zu folgenden Themen:

- Wissenschaft (Theorien, Technologien, Produkte)
- Kultur (Geschichte, Religion, Erziehung)
- Soziales (Politik, Wirtschaft, Umwelt, Schule, Familie)
- Kunst (Musik, Film, Gemälde, Computer-Grafik)
- andere Themen auf Anfrage.

Die besten Autoren werden vom 31.10.95 bis 3.11.95 zum Jugend-Gipfeltreffen nach Tokio eingeladen, um dort eine komplette Computer-Ausrüstung auf dem neuesten Stand der Technik in Empfang zu nehmen.

Hintergrund dieser Aktion ist die Idee "Alle Macht den Kids". In Deutschland wird sie von "Nippon Telegraph and Telephone Corporation" unterstützt.

Weitere Infos über die Teilnahmebedingungen sowie die "Internet Home Page Address" erfahren Sie bei der eigens dafür installierten Hotline **0228/91 47 40.**

GUSS-News

Geos User Software, Sachsen (GUSS), bietet ab sofort Hilfe bei Problemen mit Hardware-Kabeln: Wer beispielsweise ein 64NET-, Null-Modem- oder Paralleldrucker-Kabel sucht (auch mit individueller Länge) bekommt es neu hergestellt oder preiswert besorgt. Auch raffinierte Hardware-Kombinationen werden auf Wunsch konfiguriert (z.B. Btx-Modem von Drews in Verbindung mit Drucker, betrieben per Data-Switch-Schalter).

GUSS-Software läßt sich ab sofort als TSW bei *matting# abrufen. Damit kann man preiswert (ca. fünf Mark für zwei bis vier kommerzielle Programme) seine Geos-Diskothek erweitern. Letzte Meldung: "GeoCom Tips & Tricks 2" erscheint Ende August!

Infos: Geos User Software Sachsen,
Denis Döhler, Gorkistr. 18,
04347 Leipzig, Tel. 0341/23 30 180



Audio-CD-Player für C-128-User

Wer ein SCSI-CD-ROM-Laufwerk am C 128 betreibt (mit dem unverzichtbaren Umweg über eine integrierte CMD-Harddisk), kann dem Gerät ab sofort auch Töne entlocken. Möglich macht's die raffinierte Software "CDDA 128", die neueste Kreation von Achim Täge.

Das Programm läuft im 40-Zeichenmodus des C 128 - außer den Floppies 1541 bzw. 1571 braucht man dazu unbedingt eine HD von CMD, an der das CD-ROM-Laufwerk hängt und eine Maus in Port 1. Nach Installation der Software läßt sich jede beliebige Audio-CD ins Laufwerk schieben und wie auf der Stereoanlage abspielen (vorausgesetzt, Ihr Monitor ist mit entsprechenden Lautsprechern ausgerüstet).

Die Vollversion kann maximal 36 Titel einer CD zeigen und abspielen. Außerdem läßt sich per Icon nur eine bestimmte Anzahl von Musikstücken auswählen, die man auf Wunsch in einer Endlosschleife stets aufs Neue aufrufen kann.

CD 128

Die Vollversion von "CDDA 128" kostet 24,90 Mark (per Vorkasse) oder zzgl. fünf Mark Nachnahmegebühren. Sie ist exklusiv erhältlich bei:

AT. EDV Service, Zur Hotzepar 9, 42489 Wülfrath

GIG Süd e.V: Herbsttreffen

Vom 18. bis 19.11.1995 veranstaltet der Geos-Club GIG Süd e.V. ein großes Herbstreffen in Wachsenberg/Neusitz bei Rothenburg ob der Tauber.

Günstige Übernachtungsmöglichkeiten bietet eine Pension am Ort (ca. 30 Mark). Wer will, kann auch seine Familie mitbringen – die Gegend eignet sich ideal zum Wandern.

Um Voranmeldung inkl. eines gewissen Anzahlungsbetrages wird gebeten.

Infos bei: Harald von Blumenthal, Tel. 09824/8114.

"Designer's Club" - jetzt auch auf CD-ROM

Neue Ideen für Grafik-Fans und gestaltende Berufe bringt Monat für Monat der Kreativ-Service "Designer's Club" auf einer Compact-Disk heraus. Jede Ausgabe enthält saisonal bedingte Illustrationen, Rahmen- und Design-Elemente, Geschäftsgrafiken, Symbole, Silhouetten u.a. Die Grafik-Images sind im EPS-Format gespeichert, teilweise in Farbe.Dazu gibt's "Ideas & Images", ein Magazin mit Produkttips und Layoutvorschlägen. Die CD-ROM enthält zusätzlich das elektronische Mitmach-Magazin "Idea Source" mit jeder Menge nachvollziehbaren Problemlösungen.

Eine Demo-CD erhält man gegen eine Schutzgebühr von fünf Mark bei:

Vera Kopp Fachbuchversand, Postfach 1263, 63479 Bruchköbel, Tel. 06181/75057, Fax:06181/75046.

Schulfunk- und Bildungsprogramme des WDR-Hörfunks im Internet

- Seit 5.7.95 ist der WDR online: ab sofort lassen sich über "World Wide Web" des Internet aktuelle Informationen zu Hörfunkprogrammen abrufen:
- Konturen – Bildung und Wissenschaft am Nachmittag (montags bis freitags, 14.30 bis 15.00 Uhr, WDR Radio 5),
- Über den Zaun gehört - Englisch/Französisch (sonntags, 7.45 bis 8.00 Uhr und samstags, 15.45 bis 16.00 Uhr, WDR Radio 5),
- Töne, Texte, Bilder (Medienmagazin, sonntags, 13.05 bis 13.30 Uhr, WDR Radio 5),
- Schule und Hochschule (Bildungsmagazin, dienstags, 17.05 bis 17.30 Uhr, WDR Radio 5),
- Funkkolleg (sonntags, 7.00 bis 7.45 Uhr und samstags, 15.00 bis 15.45 Uhr, WDR Radio 5). Zugang zum Netz erhält man über die Homepage des WDR "http://www.wdr.com.

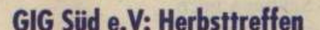
Langfristig will die Redaktion "Schulfunk und Bildungsprogramme" über reine Programm-Infos hinaus auch ausführliche Texte anbieten, z.B. den Wortlaut der Sendungen. Hierbei denkt man vor allem an den Service für Schulen, die nach dem Plan der Landesregierung Nordrhein-Westfalens flächendeckend vernetzt werden sollen.

WDR, Appellhofplatz 1, 50667 Köln, Tel. 0221/2202407, Fax: 0221/220228

Amiga-Mailbox bietet jetzt auch Geos-64-Dateien an

Obwohl ursprünglich als Amiga-Box gegründet, kann man bei der "Thunderbox" in Fürth jetzt auch Geos-Files abrufen: Speziell für die GIG Süd. e.V. hat der Sysop ein Brett mit Geos-Dateien eingerichtet. Voraussetzung ist allerdings eine ZModem-Konfiguration, die im Terminal einzustellen ist, z.B. Novaterm (sonst klappt's nicht).

Thunderbox, Fürth, Mailbox-Nr. 0911/785189



Fachbegriffe zu Disketten und Laufwerken



Kleines Floppy-Lexikon

Manche behaupten, Computer-Chinesisch sei schwerer zu erlernen als Kisuaheli.

Block Availability Map (BAM): (Blockbelegungplan in Spur 18, Sektor 0). Daraus ist ersichtlich, welche Sektoren noch frei oder benutzt sind. Außerdem informiert die BAM über Namen, ID, das Formatkennzeichen der Disk und über den Beginn des ersten Directory-Sektors.

Block: (auch Sektor) ist ein 256 Byte großer Bereich innerhalb einer Diskettenspur, der zur Datenspeicherung verwendet wird. Nur 254 Byte sind frei: die beiden ersten Speicherstellen eines Blocks enthalten Spur- und Sektornummer des logisch folgenden Blocks der aktuellen Datei (z.B. ein Basic-Programm). Automatische Verwaltung durchs Floppy-DOS.

Burst-Modus: erweiterter Floppy-Befehlssatz zur schnelleren Datenmanipulation. Funktioniert nur mit den Disk-Controllern der 1570/1571 (WD 1770) und 1581 (WD 1772).

Directory: Inhaltsverzeichnis der auf der Disk gespeicherten Dateien. Beginnt bei den Floppies 1541/1571 und 5,25-Zoll-Disketten in Spur 18, Sektor 1 (oder lt. Blockbelegungsplan), bei einer 3,5-Zoll-Disk der 1581 liegt das Verzeichnis in Spur 40. Die 1541/1571-Floppies können maximal 144 Namen ins Directory eintragen, die 1581 dagegen 296.

Double Density (DD, 2D): doppelte Dichte, bezeichnet die Speicherkapazität der Spuren. Hängt mit der Dichte der magnetischen Flußwechsel entlang des Spurumfangs und mit der Codierung beim Schreiben der Daten zusammen. Früher zeichnete man in Zweifrequenz-Schrift (FM, Frequenzy Modulatiuon) auf: Jedes Bit verbrauchte den Platz zweier Magnetflußwechsel – einfache Dichte (SD, 1D, 48 tpi). Durch geschickte Codierung konnte man den Speicherplatz pro Bit auf einen Magnetflußwechsel reduzieren: MFM-Schrift (Modified Frequenzy Modulation). Damit erhöhte sich die Spurendichte auf DD (2D), bzw. 96 tpi. Solche Disketten sind neben den C-64-Floppystationen ideal für die Laufwerke IBM-kompatibler PC/XTs.

Double Sided (DS, 2S): Diskette ist doppelseitig bespielbar. Die Bezeichnung ist irreführend, denn: Jede im Handel befindliche Diskette ist auf beiden Seiten magnetisch beschichtet, kann also nach dem Formatieren beidseitig mit Daten beschickt werden. Nur: der Hersteller vergibt mit dem Qualitätsprädikat DS (2S) quasi eine Garantie, daß es sich auf beiden Seiten problemlos speichern läßt – im Gegensatz zu SS (1S): hier übernimmt er nur die Gewährleistung für die Vorderseite.

DOS: (Disk Operating System), Betriebssystem des Diskettenlaufwerks, das Befehle des Anwenders entgegennimmt und ausführt. Bei den Floppies 1541/70/71/81 umfaßt es 32 KByte und beginnt an der internen Floppyspeicheradresse $8000 (32768). Da sich diese System-Software in einem separaten Mikrochip befindet, kann man die Daten auslesen, ändern und das neue DOS in ein EPROM brennen. Anschließend muß man die Chips austauschen.

Direktzugriffsdatei: Daten werden auf beliebig freie Sektoren einer formatierten Diskette gespeichert (DOS-Befehl Block-Write [B-W], bzw. U2). Dazu müssen Spur- und Sektornummer angegeben werden, ebenso beim Lesen per DOS-Anweisung Block-Read (B-R) bzw. U1. Der Vorteil: Da kein Eintrag im Directory erscheint, sind solche Dateien auf Disk nicht sichtbar — also nie vergessen, in welchen Spuren und Sektoren man suchen soll!

Disk-Controller (DC): Hardware (Mikrochip) in der Floppystation, der für den korrekten Laufwerksbetrieb zuständig ist. Die Floppy 1571 besitzt einen PC-Laufwerk-kompatiblen DC (Typ WD 1770), ebenso die 1581 (Typ WD 1772). Die 1541 verwendet ein Logic-Array (simpler Custom-Chip) als Disk-Controller.

Fehlerkanal: besser: Befehlskanal der Floppy. Wird beim Öffnen (OPEN-Befehl) durch die Sekundäradresse 15 eingestellt. Ermöglicht die Übergabe von Block- und Memory-Anweisungen ans Laufwerk oder die Analyse der Daten-Bytes, die z.B. einen Floppyfehler im Klartext ausgeben.

File: Fachausdruck für >Diskettendatei<. Der Name wird normalerweise auch in die BAM und ins Directory eingetragen. Ausnahme: Direktzugriffsdateien.

Group Code Recording (GCR): speziell von Commodore-Floppies verwendetes Aufzeichnungsformat für den Magnetfluß beim Beschreiben oder Lesen einer Spur.

Header: Name und ID-Kennung einer Diskette. Diese Daten sind auf Spur 18, Sektor 0 ab dem 165. Byte gespeichert.

High Density (HD): hohe Dichte. Im Vergleich zu DD-Disketten (s. Double Density) wurde der magnetische Flußwechsel erhöht (jetzt: 135 tpi). Das geht nur durch Veränderung der Diskettenbeschichtung: Bei HD-Disketten ist sie dünner, aber härter. Man sollte solche Disketten nur mit geeigneten Laufwerken bearbeiten, die bei 5,25-Zoll-Disketten 1,2 MByte und bei 3,5-Zoll-Scheiben 1,44 MByte Speicherkapazität erreichen. Die Floppies 1541/1571 sind dafür nicht geeignet: Kaufen Sie lieber keine HD-Disketten – Sie könnten (wenn nicht schon beim Formatieren!) während des Speicherns wichtiger Programme oder Daten Probleme bekommen! Unproblematisch ist dagegen die 1581.

Jobcodes: signifikante Ein-Byte-Kommandos an den Disk-Controller, die man per DOS-Anweisung Memory-Write (M-W) abschickt. Die entsprechenden Speicherstellen liegen in der Floppy-Zeropage (Adressen $00 bis $05). Nach Ausführung der Aufgabe (Job) enthalten dieselben Speicherstellen die DC-Rückmeldungen. Tabelle 3 zeigt alle Jobcodes und deren Funktionen.

Label: Etikett auf der Vorderseite mit Infos zum Disketteninhalt oder Qualitätshinweise (Spur- oder Speicherdichte usw.).

Micro-Disk: Fachbegriff für 3,5-Zoll-Disketten (z.B. für Floppy 1581, IBM-kompatible AT-, Amiga-, Atari-, Macintosh-Laufwerke). Die Bezeichung stammt noch aus der Zeit, als 8-Zoll-Disketten die Einheitsgröße waren.

Mini-Disk: Floppy-Diskette mit 5,25-Zoll Durchmesser (z.B. für die Diskettenstationen 1541, 1570 und 1571).

SYNC-Markierung: Jeder Sektor besteht aus Blockheader und dem entsprechenden Datenblock. Die Synchron (SYNC)-Markierungen bestehen aus mehreren $FF-Bytes. Findet der Schreib-Lesekopf solche Werte, kümmert er sich ums nächste Kennzeichen: Hat es den Wert $08, weiß der Disk-Controller, daß ein Blockheader folgt. Die Zahl $07 gibt an, daß nun ein 256 Byte großer Datenblock folgt. Nicht vergessen: 2 Byte muß man immer für die Spur- und Sektornummer als Zeiger auf den logisch folgenden Sektor abziehen!

Single Density: einfache Spurdichte, s. Double Density.

Single Sided (SS, 1S): Diskette kann zwar auf eigenes Risiko beidseitig bespielt werden, beim Hersteller wurde aber nur die Vorderseite geprüft (s. Double Sided).

Speeder: Soft- oder Hardware-Erweiterungen des Floppy-Betriebssystems, um Datenzugriffe (Laden und Speichern) zu beschleunigen. Hardware-Speeder sind generell mit dem Umbau der Floppystation verbunden. Damit umgeht man die langsamere serielle Datenübertragung und stellt auf Parallelbetrieb um (per zusätzlichem Kabel, z.B. am Expansionport). Software-Speeder bestehen aus einem Erweiterungsmodul, das in den Expansionport gesteckt wird (seltener aus einem Software-Programm, das man ins Computer-RAM lädt und startet, z.B."RAM-Exos" im 64'er-Sonderheft 62). Bekannte Speeder: Jiffy DOS, Rex DOS, Prologic DOS classic, Profi DOS (s. 64'er-Magazin 11/92).

Chicken

Tour de Chaos!



Mit PS-starken Boliden heizen jugendliche Rivalen über abgelegene Straßen. Wer als erster über die vereinbarte Ziellinie rast, kassiert im Normalfall den Mega-Burger oder im Bestfall die Blondine seiner Träume. Diese sogenannten Chicken-Races sind spätestens seit James Dean und Co. weitverbereitet an der Tagesordnung und mittlerweile auch hierzulande Realität. Der nicht ungefährliche Zeitvertreib hat ein Programmier-Team dazu veranlaßt, eine wesentlich sanftere Variante für den C 64 zu stricken. "Chicken" schickt den Spieler im Sprite-Truck über bildschirmgroße Pisten, wo er wahlweise gegen den Computer oder einen Mitspieler antritt. Beschleunigung und Lenkung werden recht simpel per Joystick realisiert.

Natürlich ist Wachsamkeit hinter dem Steuer gefragt, sonst macht der gegnerischen Truck den Joystick-Schumacher platt und raubt ihm eines seiner drei Spielerleben. Haben die beiden Oponenten genügend Runden, wechselt das Szenario. Das Spiel verstreut in den Leveln per Zufall Extras. Sie helfen dem Raser zu höheren Ehren und rüsten sein Vehikel mächtig auf.



Als Chicken-Pilot hat der Spieler durch die träge Joystick-Steuerung auf den verwinkelten Rennstrecken kaum rechten Spaß

Leider kann das Game, von der recht netten Grafik mal abgesehen, kaum überzeugen. Die Steuerung ist viel zu träge und die haarsträubenden Sounds verderben den Spaß endgültig. Es kommt einfach keine Stimmung auf und es steht ganz klar fest: die Idee, die Kids mit dem Bleifuß an den C 64 zu locken, ging hier gründlich in die Hose! Da hole ich lieber, das zwar etwas ältere, aber rasantere "Super Cars" aus der Garage und drehe da meine Runden.

Jörn-Erik Burkert

***Info:** Performance Periphals Europe, Michael Renz, Holzweg 12, 53332 Bornheim, Tel./Fax: 02227/3221*



Slaterman

Nach den Sternen greifen

Slaterman ist ein kleines roboterähnliches Kerlchen, das ganz heiß auf Sterne ist. Es durchwandert dazu horizontal scrollende Spielstufen und sammelt die Objekte seiner Begierde ein. Unterwegs schnappen Mäuse, Vögel und andere Viecher nach dem Helden und wollen ihm eines seiner drei kostbaren Leben berauben. Da helfen nur gezielte Sprünge oder Schüsse aus der Laserwaffe. Diese zerlegen ruckzuck die Widersacher in Staub und Asche. Zu allem Übel steht Slaterman bei seiner Jagd nach den geliebten Sternen auch noch unter Zeitdruck. Verrinnt das Limit, wird ihm ein Leben von seinem Konto abgezogen.

Die Threshold-Timsoft-Koproduktion sorgt spätestens nach dem zehnten Level für nachhaltiges Gähnen beim Spieler, denn der Levelaufbau zeigt sich ähnlich abwechslungsreich wie der Kleinanzeigenteil einer Tageszeitung. Verschiedene Abschnitte wiederholen sich in regelmäßigen Abständen und die eigentlich recht nette Hintergrundgrafik verliert auch, weil sie leider auch immer die gleiche bleibt. Hätten sich die Designer sich nicht nur auf das Einsammeln der Sterne und dem Ausweichen bzw. Abballern beschränkt, wäre das Spiel hitverdächtig – einige Fahrstühle, Fallen, Geheimräume und Boni hätten da gut getan. Außerdem wäre nach wenigstens jedem zehnten Level ein neues Szenario mit Extra-Grafik angebracht. Die ungenau arbeitende Kollisionsabfrage sorgt außerdem für Frust. Ballert Slaterman seine Gegner um, reißt die Explosion manchmal den Helden urplötzlich in den Tod.

Einige Extras mehr, Feinschliff beim Gameplay und ein bißchen mehr Abwechslung würden das Spiel in die Oberliga kicken. So bleibt leider nur ein mittlerer Rang übrig. *Jörn-Erik Burkert*



Der kleine Held schaut auf die Objekte seiner Begierde – die Sterne werden aber von heimtückischen Gegnern umschwärmt

***Info:** Treshold-Productions, 17730 15th Ave NE Suite #229, Seattle, WA 981 55, USA, Telefon: ++1/206/361 1332, Internet: tpinfo@eskimo.com*

Slatermann

64'er WERTUNG **6** von 10

- Spielidee
- Grafik
- Sound

Schwierigkeit **mittel**

Software-Perlen für den kleinen Geldbeutel

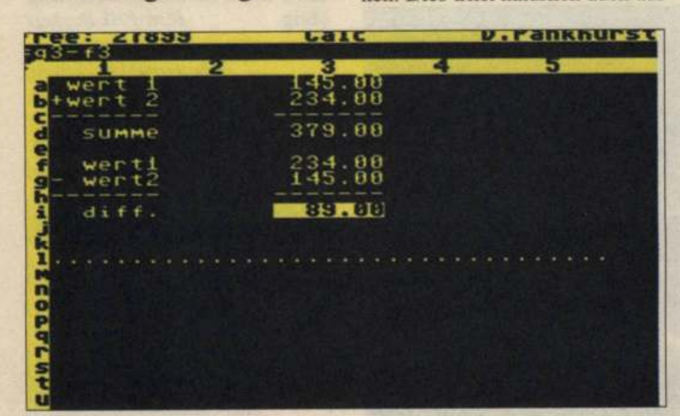
Viele gute C-64-Programme kommen aus den Bereichen Public Domain und Shareware. Wir haben ein wenig im Sofware-Angebot gestöbert und aus der Vielfalt einige Perlen herausgepickt.

Seinen Erfolg hat der C 64 letztlich auch dem breiten Software-Angebot zu verdanken. Dies trifft natürlich auch auf die Bereiche Public Domain und Shareware zu. Und das bedeutet Geld sparen.

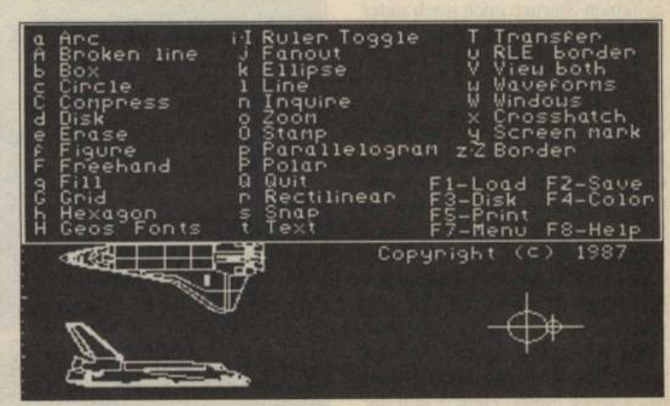
Mit ein bißchen Geduld beim Studium der Händler-Kataloge, findet man sehr oft wahre Schätze.

Anwender-Lösungen

In dieser Rubrik deckt das Angebot fast jeden Bereich ab. Egal ob Textverarbeitung (z.B. Kwik-Write), Dateiverwaltung oder



Mit der kanadischen Tabellenkalkulation "Calc" von David Pankhurst haben Sie Ihr Haushaltsbuch, Abrechnungen oder die Bankauszüge schnell im Griff



Mit "CAD 4.0"lassen sich die unterschiedlichsten Objekte konstruieren - das Programm aus dem Jahre 1987 ist leider etwas langsam und reagiert auf Floppyspeeder allergisch

Lernprogramm - für fast jeden Bereich gibt es ein Programm. Mit "Calc" haben Sie Ihre Kalkulationen wie die Profis im Griff. Konstrukteure können mit "CAD 4.0" Objekte entwerfen und werden dabei durch viele Befehle unterstützt. Leider steht das Floppy-intensive Werkzeug mit Speedern auf Kriegsfuß. Lernprogramme helfen Ihnen beim Büffeln von Fremdsprachen oder bringen Ihre Kenntnisse in Mathematik oder Physik auf Vordermann. Mit den unzähligen Packern, Kopierprogrammen und andern Disk-Utillities haben Sie rasch Ihre Disketten-Sammlung in Ordnung. DFÜ-Fans wiederum stürzen sich auf so leistungsfähige Programme wie "Novaterm".

Musik, Grafiken und Szene-Demos en masse

Für DTP-Freunde und Grafik-Freaks stellt sich sehr oft die Frage nach geeigneten Motiven, Zeichensätzen oder Cliparts. Hier sind PD- und Shareware-Sammlungen eine schier unerschöpfliche Fundgrube. Egal ob Geos, Printfox oder Koala-Grafiken - fragen Sie Ihren Händler, er hat was! Die Grafiken sind Handmade oder sehr oft auch gescannte Motive. Passende Bildviewer und Konverter finden Sie in jeder guten PD-Shareware-Sammlung. Für Computer-Komponisten hat der Public-Domain-Pool unzählige Musik- und Drum-Programme und passende Sounddateien auf Lager.

Die C-64-Demo-Szene nimmt einen breiten Raum im Bereich Public-Domain ein. Viele Anbieter haben umfangreiche Serien mit Szene-Demos oder Disk-Mags in ihrem Angebot. Sie warten mit Effekten, Mega-Sounds und Grafiken vom Feinsten auf.

Spiele-Stuff

Im Bereich Games tummeln sich viele Titel. Leider erweist sich so mancher Kandidat als faules Ei mit mieser Grafik und Programm-Fehlern. Das Adventure "Harak Iri" beweist, daß es auch anderes geht! Witzige Texte und ein leistunsgfähiger Parser sorgen für viel Spielspaß bei der Jagd nach dem Bösewicht "Kart Offel" und seinem Spießgesellen "Pant Offel".

Freunde der Wirtschaftsimulation werden bei "Kaiser II" recht schnell Raum und Zeit vergessen. Als Edelmann im Mittelalter muß man hier für das Wohlergehen seiner Länderein sorgen und sich gegen andere Landesfürsten durchsetzen. Auch nach zehn Jahren , hat das Kultspiel "Boulderdash" noch nicht ausgedient. Dank der "Rockford-PD-Serie", lebt der diamantenhungrige Held Rockford weiter. Tools und ein Konstruktions-Kit animieren zum Eigenbau von Boulderdash-Leveln und Caves. Mit den Intro-Designern kann man Vorspänne für neue Boulder-Dash-Spiele in Eigenregie zusammenbasteln.

Außerdem findet man unzählige neue Boulderdash-Caves, News zum Thema "Boulderdash" und noch andere nützliche PD-Programme auf den Disketten.

Programme für den C 128

Besitzer eines C 128 betreiben den Computer sehr oft nur als C 64 und vernachlässigen den zweiten Betriebsmodus. Speziell für den C 128 gibt es einige sehr interessante Programme, die vor allem die 80-Zeichen-Darstellung oder das Betriebssystem CP/M nutzen.

Hervorzuheben sind dabei Utilities wie "CS-DOS" (unterstützt die Arbeit mit dem Betriebssystem) oder das DFÜ-Programm "DesTerm". Mit "ACE 128" verpassen Sie ihrem C 128 eine Oberfläche à la Unix. Außerdem finden Sie im PD- und Shareware-Pool unzählige Programme für den CP/M-Betriebsmodus des C 128.

Wo gibt's die Software?

Um gute Shareware- oder Public-Domain-Programme zu bekommen, wenden Sie sich am besten an einen entsprechenden Händler. Die Adressen finden Sie im 64'er-Magazin. Eine weitere Möglichkeit ist die Datenreise per Modem. In C-64-Mailboxen (z.B. DreamBeam Tel.: 08682/9779) findet man sehr oft ein umfangreiches Angebot. Ebenso in BTX (z. B. Brotkastencorner - *matting#).

Jörn-Erik Burkert

Info: *Stonysoft, Beethovenstr.1 , 87727 Babenhausen, Tel.: 08333 1275, Fax: 08333 7044*
Data House, Harteshäuser Str. 67, 34130 Kassel, Tel.: 0561 68012, Fax: 0561 68405
Evolution, Berliner Str. 55, 24345 Eckernförde, Tel.: 04351/5374

Die Brotkasten-CD

Viele MBytes Shareware und Public Domain für C 64/128 finden Sie auf Deutschlands zweiter C-64-CD-ROM! Insgesamt 1111 Disketten (voll mit freikopierbarer Software) soll der Silberling bergen. Das Spektrum reicht von Anwendungen und Spielen, über Grafiken und Geos bis hin zu C-128-Programmen.
Außerdem bietet die CD noch einige Extras:
- zehn Vollversionen kommerzieller Geos-Programme von GUSS und Performance Periphals (u.a. Geos-Diskeditor, PP-Collection, NLQ-Print, Makroform)
- Novaterm (Besitzer der CD bezahlen fünf Mark weniger bei der Registrierung)
- CD-Commander-128-Testversion (zum Anschluß von SCSI-CD-ROM-Laufwerken an CMD-Festplatten)
- Testversion 64net (benutzen Sie Festplatten eines PC mit dem C 64)
- PC64 - Shareware-Version des C-64-Emulators für PC von Wolfgang Lorenz
- C64S - C-64-Emulator für PC von Miha Peternel
- A64 - C-64-Emulator für den Amiga
- Geos-Patch für C-64-Emulator (Geos läuft auf PC!)
- Hilfstools zur Bearbeitung von CD-ROM-Dateien (Trans64, X1541...)
- Emulatoren für CP/M, Atari, Spektrum, ZX81
- Kult-Bilder aus der Commodore-Geschichte, Texte aus dem Internet

So verwenden Sie die CD-ROM:

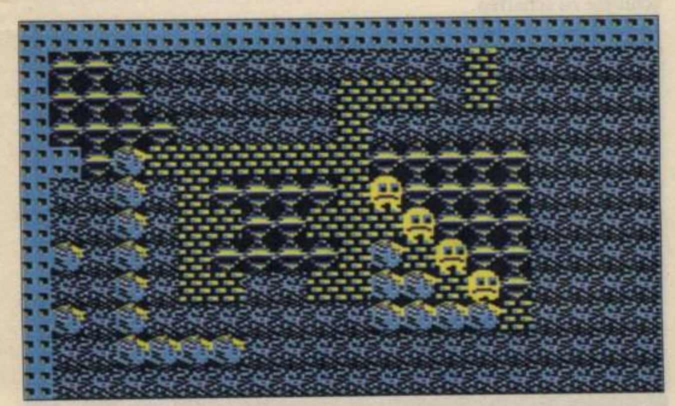
C 64/128
- der "CD Commander" läßt direkten Zugriff auf die Daten eines SCSI-CD-ROM-Laufwerks zu. Das Laufwerk wird an einer CMD-Festplatte angeschlossen.
- mit "64net" nutzen Sie Festplatten und das CD-ROM-Laufwerk eines PCs (ab XT)

PC
- mit den registrierten Versionen von PC64 oder C64S können Sie eine Floppy 1541 an Ihren PC anschließen
- mit den Utilities "X1541", "StarCommander" oder "Trans64" (alle auf der CD) kopieren Sie die Daten per Kabel zwischen PC und C 64

Amiga
- die registrierte Version des Emulators A64 läßt den Anschluß einer Floppy 1541 zu.

Auf der PC-Seite bietet die CD eine komfortable Windows-Oberfläche, die u.a. Dateien entpackt, Texte bzw. Grafiken anzeigt und den Inhalt von C-64-Diskimages (D64) sichtbar macht. Der Preis der CD soll 49 Mark betragen.

Herstellung: *Firma Lutz Hillmann, CD-ROM-Produktion, Steinstr. 3/503, 01069 Dresden*
Vertrieb auch: *Performance Periphals, Stonysoft,* ***BTX-Direktbestellung:*** **matting#*



Boulderdash for ever! Dank der Rockford-Reihe lebt das Kultspiel weiter - Konstruktion-Kit, Intro-Designer und Linker machen die ganz private Boulderdash-Eigenkreation zum Kinderspiel

Der zweite Teil von Kaiser ist eine gelungen Wirtschaftssimulation - als Blaublütiger im Mittelalter, heißt es handeln, schachern, wirtschaften und abkassieren

Die Geos-Story

Gestern, heute, morgen...

Die beliebte C-64-Benutzer-Oberfläche Geos wurde vor knapp zehn Jahren aus der Taufe gehoben. Wir haben wißbegierig im Archiv gewühlt und präsentieren Ihnen die Geschichte eines der erfolgreichsten Software-Pakete für den C 64.

Als im Herbst 1983 vier ehemaligen Mitarbeiter der Firmen Mattel und Imagic "Berkeley Softworks" gründeten, gab's noch lange keinen Gedanken an die Benutzeroberfläche Geos für den C 64.

Die neue Firma brauchte Geld, übernahm zunächst einige Aufträge von Fremdunternehmen und wirkte als Hard- und Software-Entwickler bzw. Berater.

In dieser Zeit entstand z.B. ein Mini-Computer und es wurden Auftragsprojekte für die Firmen Sega und Activision realisiert.

Außerdem schrieben die Entwickler von Berkeley Softworks einige Versionen des Druckporgamms "Printmaster".

Die ersten eigenen Produkte stellte Berkeley Softworks erst auf der CES 1985 vor: Das 15-köpfige Entwickler-Team brachte Development-Tools für den C 64 auf den Markt. Diese erleichterten durch ihre Abstimmung und Normung bei Hard- und Software die Entwicklung von C-64-Programmen ungemein.

Im August 1985 wurde der Grundstein für Geos gelegt. Firmengründer Brian Dougherty verwirklichte damit eine Idee, die ihn schon länger beschäftigte.

Die ersten Schritte - Geos entsteht

Die Entwicklung von Geos erfolgte auf UNIX-Computern (Typ Integrated Solutions Supermicro), wobei jeder Programmierer seinen eigenen Terminal zur Verfügung hatte. Diese Methode gewährleistete effektives Zusammenarbeiten: sie ermöglichte nämlich jedem Entwickler den Zugriff auf alle Routinen und Applikation die innerhalb des Geos-Teams geschrieben wurden, da die Programme und Quelltexte auf der Festplatte des Zentralrechners der Firma gespeichert wurden.

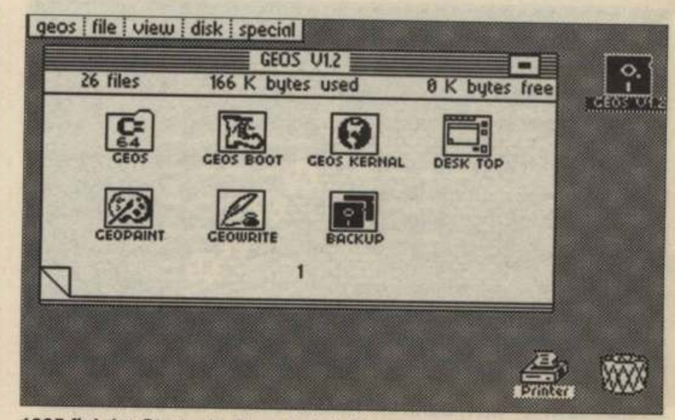
Neben den Terminals des Großrechners waren natürlich immer ein C 64 und eine Floppy 1541 plaziert. Der Computer war durch einen "In Circuit Emulator" (ICE) mit der UNIX-Maschine verbunden. Diese Hardware wurde schon 1984 durch Berkeley Softworks entwickelt und stellte den direkten Draht zwischen Entwicklerplattform und C-64-Prozessor dar.

Mit dem ICE konnten die Entwickler den Programm-Code blitzschnell in den Speicher des C 64 übertragen und das Programm beim Ablauf auf dem Bildschirm der Entwickler-Plattform überwachen und eventuelle Fehler aufspüren. Diese zusätzliche Hardware kostete damals stolze 5000 US-Dollar pro Stück, ist aber heute nicht mehr auf dem Markt.

In den "Gründerjahren" waren viele Programmierer noch Studenten, die die Entwicklung von Geos und den dazugehörigen Applicationen nur als Nebentätigkeit ansahen. Da aber die Entwicklung eines Software-Pakets wie Geos recht viel Zeit in Anspruch nimmt, mußte so mancher Entwickler quasi Urlaub nehmen, um sein Studium fortzusetzen und die Abschlüsse zu schaffen.



Die C-64-Version des Druckprogramms "Printmaster" wurde von Berkeley Softworks entwicklet



1985 fiel der Startschuß zur Entwicklung von Geos - Version 1.2 wurde kurze Zeit später der Computer-Gemeinde vorgestellt



Die ersten Applikationen

Geos war noch beim Projektstart als Benutzeroberfläche wie bei einem Apple Macintosh geplant. Natürlich lebt ein Betriebssystem nur, wenn Anwendungen dafür existieren. Deshalb wurden die Textverarbeitung *GeoWrite* und das Malprogramm *GeoPaint* entwickelt. Vor allem die Möglichkeiten bei der Text-Bearbeitung und die Bildschirm-Darstellung von *GeoWrite* waren ein Paukenschlag. Nach und nach folgten weitere Applikationen, wie z.B. *GeoCalc* (Tabellenkalkulation), *GeoFile* (Dateiverwaltung) und *GeoSpell* (Rechtschreibprüfung für GeoWrite). Später kamen *GeoChart* (Programm für Säulen-Diagramme usw.) und das DTP-Programm *GeoPublish* hinzu. Im Rahmen eines 64'er-Programmier-Wettbwerbs entstand 1989 das DFÜ-Programm *GeoTerm*.

Nach den Versionen 1.3 und 2.0 folgte "Geos 2.5" mit integriertem TopDesk und einigen Zusatztools

Neue Geos-Versionen

Im Sommer 1987 erschien "Geos 1.3" in den USA und kurze Zeit später auch in Deutschland. Die deutsche Version beherrschte endlich Umlaute und kannte sogar das "ß". Mit der Version 1.3 machten die Geos-Entwickler einen gewaltigen Schritt nach vorn. Zum ersten Mal arbeitete Geos mit den Commodore-RAM-Erweiterungen 1750 und 1764 zusammen. 166 Kbyte wurden durch die Benutzeroberfläche in eine RAM-Floppy verwandelt, die wie eine VC 1541 arbeitet. Der Rest wurde als Shadow-RAM genutzt, in dem häufig gebrauchte Programmteile zwischengespeichert wurden. Außerdem wurde die neue Commodore-Floppy 1581 bei der Entwicklung einbezogen.

Parallel zur Version 1.3 arbeitete man bei Berkeley an einer speziellen Geos-Variante für den C 128. Sie debütierte zum Jahreswechsel 1987/88 präsentiert und nutzte den 80-Zeichen-Schirm des C 128 aus.

Geos 2.0 und 2.5 im Vormarsch

Ende 1988 kündigten die Entwickler von Berkeley Softworks die Version 2.0 von Geos an und im Frühjahr 1989 schon konnten alle Freaks hierzulande die deutsche Version der grafischen Benutzeroberfläche erwerben.

Am Betriebssystem hatten die Programmierer gründlich gefeilt und unzählige Bugs der Version 1.3 behoben. Probleme bei der Zusammenarbeit mit der Floppy 1581 waren nun Geschichte.

Datei-Symbolen (Pictogramme) konnten unterschiedliche Farben zugewiesen werden und die integrierte Textverarbeitung *GeoWrite* wurde um viele nützliche Features ergänzt.

1993 ist es dann soweit: die Geos-Version 2.5 erblickt das Licht der Welt und erschien im Markt & Technik Verlag. Ein überarbeiteter Desktop, der "TopDesk", ist die größte Neuerung. Bis zu vier

Directory-Fenster darf der Geos-Fan jetzt öffnen. Durch einfaches Verschieben der Icons zwischen den Directory-Fenstern werden nun Dateien kopiert. Dabei unterstützt Geos den User durch automatische Erkennung der Disks und vermeidet Bedienungsfehler.



Der typische Arbeitsplatz eines Geos-Entwicklers bei Berkeley Softworks - neben dem UNIX-Terminal steht ein C 64 mit Floppy und ein ICE (In Circuit Emulator)



Der "In Circuit Emulator" (ICE) verbindet die Entwicklerplattform mit dem C 64

Außerdem lassen sich mit der Version 2.5 von Geos endlich auch die Unterverzeichnisse (in Form von Ordnern) bearbeiten und erzeugen.

Die Größe der TopDesk-Fenster ist veränderbar und der angezeigte Ausschnitt läßt sich im Window in alle vier Richtungen scrollen. Einige sehr nützliche



Die Geos-Väter Mitte der 80er Jahre: Viele Entwickler bei Berkeley Softworks waren in der Gründungszeit Studenten - für Prüfungen und Examen mußten sie sich "Urlaub" nehmen

Tools runden die Palette des Programmpakets ab.

RAM und ROM für Geos

Zugeben, der Speicher des C 64 ist für ein gewaltiges Projekt wie Geos ein wenig knapp bemessen. Zwar wurden seit der Version 1.3 die Commodore-RAM-Erweiterungen unterstützt, doch das sollte für Berkeley noch nicht der Weisheit letzter Schluß sein. Deshalb entwickelten die Kalifornier 1991 eine Speicherweiterung mit 512 KByte. Mit *GeoRAM* wird der User seit dem mit ausreichendem Speicherplatz beim Geos-Einsatz versorgt und hat einen passablen Ersatz für die nicht mehr produzierten Commodore-Expansions.

Mit *GeoROM* läßt sich seit 1991 die Oberfläche (nur für die C-64-Version von Geos) direkt aus einem Eprom booten. Das erspart natürlich lästige Warterei und nervende Diskettenwechsel beim Programmstart.

Geos nicht nur für Anwender

Die vielen Anwendungen für Geos bescherten dem System schnell eine große Fan-Gemeinde. Viele User entwickelten schon bald den Wunsch, selbst Programme für Geos zu schreiben. Dieser Tatsache wurde Rechnung getragen und *GeoBasic* und der *Geo-Programmer* auf den Markt gebracht.

Während *GeoBasic* sich mit zu vielen Kinderkrankheiten herumschlagen mußte, setzte sich die Maschinensprache-Programmierung mit dem *GeoProgrammer* recht schnell durch. Dieses Tool wurde später durch den *MegaAssembler* ergänzt, der durch ein umfangreiches deutsches Handbuch glänzte.

Bei beiden Assemblern werden die Quelltexte mit *GeoWrite* geschrieben und später in Maschinensprache übersetzt. Ein Linker macht die Dateien lauffähig.

Eine Alternative zu Basic und Maschinensprache ist das Entwicklungs-Paket *GeoCom* von Falk Rehwagen. Zwar ist das Programmier-Tool an Basic angelehnt, hat aber auch zahlreiche Elemente aus anderen Hochsprachen integriert.

Auf die Zeilennummern wurde ganz verzichtet und die ca. 250 Befehle bieten einen starken Leistungsumfang bei der Entwicklung von Software.

Spitzen-Software ohne Ende

In den letzten Jahren hat sich das Software-Angebot für Geos stark erweitert. Die meisten Entwicklungen kommen aus den USA und Deutschland. Die Palette reicht von Anwendungen, über Entertainment bis zu Spielen.

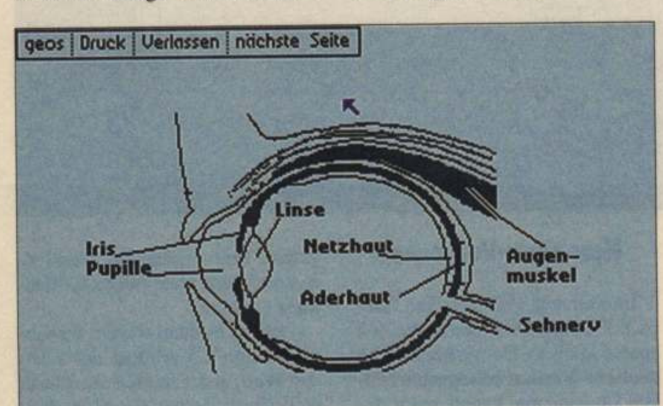
Leider hat Berkeley Softworks die Geos-Entwicklung für den C 64 eingestellt und sich voll auf den PC gestürzt. Trotzdem lebt die C-64-Geos-Szene. Titel wie *Geo-Tech* oder das Multiplayer-Spiel *Trade & War* zeigen, welche tollen Geos-Programme weiterhin auf den Markt kommen.

Die Einbindung von Geos in *64net* macht nun auch den Zugriff auf die Laufwerke eines PCs möglich. Ein Patch erlaubt sogar die Geos-Nutzung auf dem C-64-Emulator PC64.

Eine neue Geos-Version haben einige Entwickler schon in der Mache und ein BTX-Decoder für die grafische Benutzeroberfläche ist in Vorbereitung. Die einfache Bedienung und die vielen Features haben das System etabliert und erfolgreich gemacht.

Jörn-Erik Burkert

***Info:** Performance Peripherals Europe, Michael Renz, Holzweg 12, 533332 Bornheim, Tel. und Fax: 02227/3221*



Geos bietet auf vielen Terrains Software - neben Anwendungen, findet man nun auch Lernsoftware wie "GeoMensch"



Eine der bekanntesten Applikationen für GeoWrite - diese Version läuft auf dem C-64-Emulator für PC

Geo(s)logisch wertvoll

Public-Domain- und Shareware-Programme entpuppen sich immer öfter als nützliche Helfer bei der Arbeit mit der Benutzeroberfläche Geos von Berkeley Softworks. Einige nützliche Tools, die dem Geos-User die Arbeit erleichtern, haben wir für Sie herausgesucht.

Trotz aller Unkenrufe und der Konkurrenz auf großen Computern - die Geos-Szene auf dem C 64 und C 128 lebt! Die unzähligen Aktivitäten der Geos-Freaks spiegeln sich in der Software-Vielfalt in den Bereichen Public Domain und Shareware wider.

Neue Systemschrift: ChangeBSW

Mit diesem aus Amerika stammenden Hilfs-Programm können Sie die im Geos-System fest integrierten Fonts "BSW" bzw. "BSW 128" durch andere Zeichensätze von Diskette recht einfach ersetzen.

Als selbststartendes Programm muß es mit zwei Fonts, einer für 40 Zeichen und einer für 80 Zeichen (Geos 128) auf die Geos-Boot-Diskette kopiert werden. Die Version 2.0 dieses Programms wurde der deutschen Geos-Version angepaßt. Im Lieferumfang sind einige Fonts, von denen sogar zwei mit deutschen Umlauten versehen wurden.

Ein kleines Problem sollte aber nicht verschwiegen werden: die Shortcuts (mit Commodore-Taste) machen ein wenig Ärger, denn sie werden nicht mehr korrekt in den Pull-Down-Menüs dargestellt.

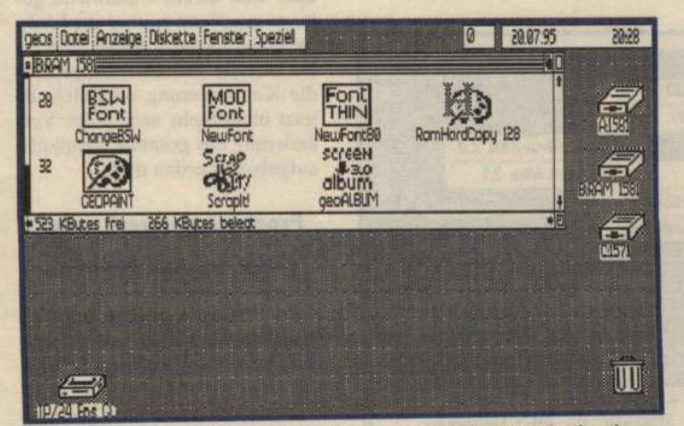
Die Vorteile lassen aber über diesen Verlust hinwegsehen - GeoCalc druckte bisher beispielsweise im Grafikmodus nur mit dem BSW-Font.

Jetzt besteht die Möglichkeit, einen anderen Zeichensatz zu benutzen oder der Desktop-Oberfläche ein neues Outfit zu verpassen.

Zeichensatz tauschen: Font_it!

Da GeoWrite im Schrift-Menü nur die ersten sieben Fonts auf Diskette anzeigt, kommt es manchmal zu Problemen. Wer kennt das nicht: man schreibt einen Brief mit GeoWrite, doch der gerade benötigte Font liegt an achter Stelle oder noch weiter hinten auf Diskette. Was bleibt: GeoWrite verlassen, die Reihenfolge der Fonts auf der Arbeitsdiskette neu ordnen und GeoWrite neu starten.

Mit dem Programm "Font_it!" kann diese umständliche Prozedur jetzt für das deutsche Geos 64 endgültig zu den Akten gelegt werden: das Programm läßt sich als Hilfsmittel direkt aus der Textverarbeitung heraus über das Geos-Menü starten. Nun besteht die Möglichkeit, einen oder mehrere Fonts, die weiter hinten auf Diskette stehen, mit einem der ersten sieben Fonts auf Disk auszutauschen. Ab sofort läßt sich dieser Font uneingeschränkt in GeoWrite benutzen.



Der TopDesk 128 mit neuer Schrift - der neue Font wurde durch das Utillity "ChangeBSW" installiert. Es arbeitet sowohl mit der Version für den C 64, als auch für C 128

Druckertreiber: Stylus NLQ

Hier handelt es sich um ein System von 16 Druckertreibern für den NLQ-Modus der Tintenstrahler *EPSON Stylus 800* und *EPSON LQ570*. Die Treiber liegen für parallelen und seriellen Druckeranschluß in verschiedenen Varianten vor.

Sie arbeiten mit Steuerzeichen im Text. Dadurch kann das Leistungsangebot dieser Drucker im NLQ-Modus unter Geos 64/128 voll genutzt werden.

Neben den Schrift- und Stilvarianten, die GeoWrite zur Verfügung stellt, kommen eine Reihe weiterer Druck-Features zum Einsatz:

- alle im Drucker vorhandenen Schriftarten und -Größen
- unterstreichen, überstreichen und durchstreichen in verschiedenen Variationen
- Shadow (ähnlich Kontur)
- Druck von ASCII-Zeichen außerhalb des in Geos möglichen Bereichs von 32-127.

Das besondere an diesen Druckertreibern ist, daß sie nicht nur mit Stylus-Druckern, sondern mit allen Druckern zusammenarbeiten, die ESC P2 kompatibel sind. Dazu gehört unter anderm auch der 24-Nadler "EPSON LQ-100".

Verbesserte Version: Patch Updater

Für effektives Arbeiten unter Geos empfiehlt sich, das Programm auf eine RAM-Disk zu kopieren. Nach Ende der Sitzung müssen die Dateien wieder auf Diskette gespeichert werden.

In der Regel kopiert man die geänderten Dateien im Desktop auf eine Diskette. Da kann es schon mal vorkommen, daß eine Datei bei diesem Vorgang vergessen wird oder man kopiert eine Datei, die gar nicht verändert wurde.

Um diesen Prozeß zu automatisieren, wurde das Programm "Updater" geschrieben. Die aktuelle Updater-Version ist V2.0. Sie ist kommerzielle Software

und wird u.a. auf der Diskette "Spezial #1" vom Geos User Club vertrieben. Das vorliegende Patch-Programm erzeugt die Version 2.1 von Updater. Daraus ergeben sich Verbesserungen:

- Updater unterstützt jetzt auch Topdesk-Ordner
- die Kopierroutine in Nativ-Mode-Partitionen von CMD-Geräten (z.B. Festplatten) arbeitet jetzt fehlerfrei
- die Routine zur Laufwerkstyp-Erkennung funktioniert nun mit alle Drives und die richtigen Kennungen des GATEWAY werden gecheckt
- die Laufwerks-Icons im Programm werden geändert.

Damit sind wohl alle Wünsche erfüllt und das Programm Updater ist in der Version 2.1 endlich ausgereift und funktioneller als sein Vorgänger.

Kommunikations-Tool: BTX-Convert 2.1 (L)

Wer mit dem Drews BTX-Decoder und der Erweiterung BTX-Extra von Wolfgang Grimm arbeitet und die mit der Funktionstaste *<F6>* gespeicherten BTX-Seiten komfortabel in GeoWrite weiterverarbeiten möchte, erhält mit diesem Konverter wertvolle Hilfe.

Es erzeugt aus den BTX-Seiten ein GeoWrite-Dokument, in das bis zu 60 BTX-Seiten aufgenommen werden können. Dabei lassen sich das Format (Write Image V2.0 oder V2.1) und der Font (BSW, BSW 128 bzw. Commodore_GE) wählen.

Außerdem unterdrückt der Konverter auf Wunsch die Zeilen 1 und/oder 24 einer BTX-Seite, die nur Statusmeldungen enthalten und bei der Verarbeitung der Textinformationen sehr oft stören. Eine etwas eingeschränkte Version des Programms wird jetzt als Shareware vertrieben. Gegenüber der Version 2.0 wurde die Menüsteuerung ein wenig geändert.

Alle Punkte können jetzt nur noch nach oben verlassen werden. Von einem Untermenü gelangt man jetzt immer ins nächste übergeordnete Menü. Das erleichert die Voreinstellungen für die Konvertierung erheblich, da jetzt nicht mehr nach jeder Veränderung das gesamte Menü neu aufgebaut werden muß.

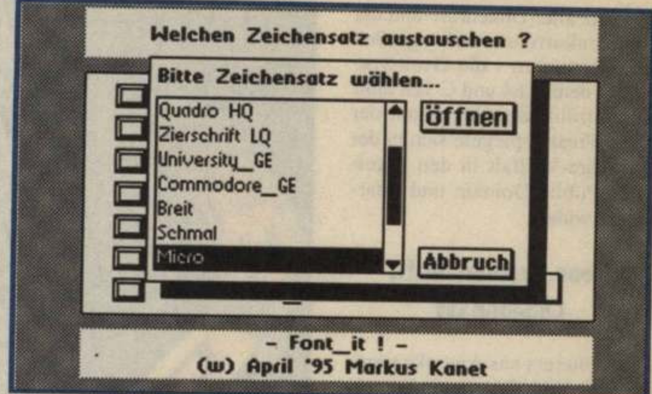
Programmierwerkzeuge: Forth 1.2 und geoCOPE

Für Programmierer hat der PD/Shareware-Versender Stonysoft zwei Geos-Pakete im Softwareangebot:

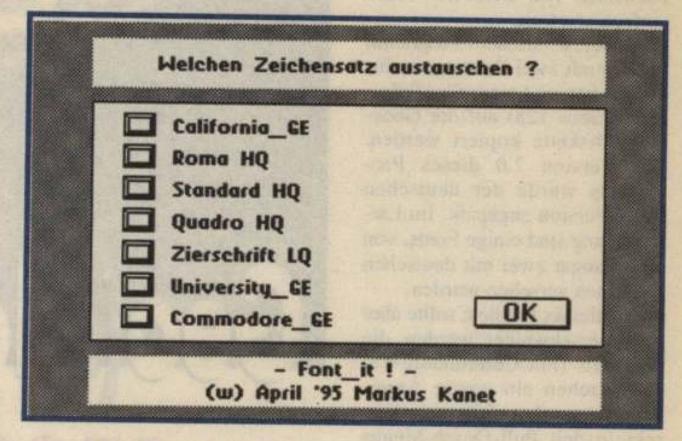
Die Programmier-Sprache *Forth* in der Version *1.2* und das Entwicklungspaket *geoCOPE* (Assembler und Editor).

Beide Entwicklungs-Programme sind mit englischer Anleitung und diversen Tools zu haben. Auf der Diskette mit Forth für Geos finden Sie außerdem den Monitor *GeosMon, Convert 2.5* und *Geosbusters*.

Werner Weicht/lb



Mit dem Tool "Font-It" läßt sich in GeoWrite blitzschnell die Zeichensatz-Reihenfolge ändern - das Filer-Menü unterstütz den User dabei und macht die Auswahl zum Kinderspiel



"Font_It" tauscht die Einträge in der Geos-Font-Liste und unterstütz die Arbeit mit GeoWrite sehr intensiv

Werner Weicht

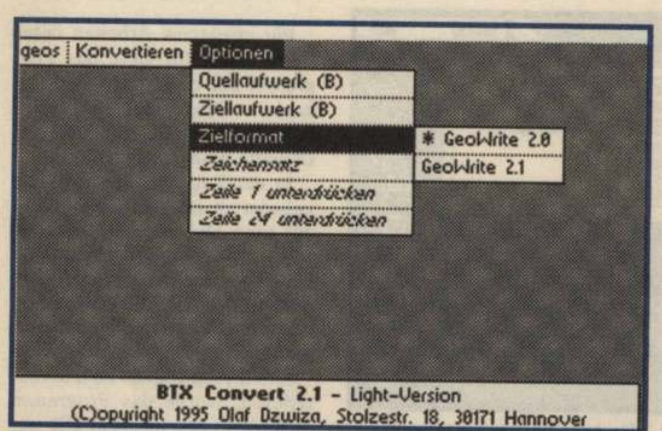
Die Druckqualität unter Geos mit dem Treiber "Stylus NLQ" kann sich durchaus sehen lassen - die 16 Treiber unterstützen u.a. Printer die ESC-P2-kompatibel sind

Patch UPDATER 2.0 (c) '95 L. Eckhardt

Dieses Patch ändert die Kopierroutine, um kompatibel mit CMD-Nativ-Partitions zu werden. Die Erkennung des Laufwerkstypes wird in Bezug auf die GATEWAY Treiber korrigiert und das HD-Icon wird verändert.

UPDATER patchen !

Das Programm "Updater" wird durch eine Patch verändert - die Software generiert Version 2.1. - sie ist leistungsfähiger und kompatibler als ihr Vorgänger



Texte die mit dem Drerws-BTX-Decoder und BTX-Extra empfangen wurden, lassen sich mit BTX-Convert ins GeoWrite-Format wandeln

Wo gibt's die Programme?

Die vorgestellten PD/Shareware-Programme liegen als Telesoftware in vielen Mailboxen und zum Teil auch in BTX (z.B. *"geos#* oder *"matting#*) zum Download bereit. Einfach das Modem anwerfen, Terminalprogramm aktivieren und sich in eine Box einloggen...

Außerdem sind sie in der Geothek des Geos User Club und bestimmt auch bei anderen PD/Shareware-Händlern zu bekommen.

Speziell Stonysoft in Babenhausen hat eine umfangreiche Geos-Serie in seinem PD/Shareware-Programm. Erkundigen Sie sich nach dem aktuellen Produkt-Katalog.

Geos Userclub, Moerser Str.11, 46286 Dorsten, Tel.+Fax: 02866-376

Stonysoft, Beethovenstr.1, 87727 Babenhausen, Tel. 08333/ 1275, Fax: 08333/ 7044

Trade & War



Der Kampf um die reichen Jagdgründe des Universums ist eröffnet! Mit uns können Sie bei diesem Abenteuer dabei zu sein. Mit dem Geos-Frontend auf unserer Diskette zum Heft, fighten Sie bei "Trade & War" mit. Viel Erfolg!

Das Spiel "Trade & War" ist ein Multiplayer-Spiel, an dem bis zu 25 Personen teilnehmen. Nach jeder Spielrunde schicken die Teilnehmer ihren Zug per Modem zum Host (Mailbox). Der Spielleiter sammelt alle Dateien und startet, wenn alles eingetroffen ist, ein Auswertungsprogramm. Dieses läuft aus Geschwindigkeitsgründen auf einem PC. Nach der Bearbeitung der Datenfiles berechnet der PC die neue Spielsituation und generiert neue Ergebnis-Files, die die Spieler für die nächste Runde aufs Auge gedrückt bekommen.

Spielvoraussetzungen

Bevor es in den Kampf um die Herrschaft im Weltraum geht, muß Ihr Computer-System erst mal einige Voraussetzunegn erfüllen: Zunächst benötigen Sie ein Modem, DFÜ-Software zum Austausch der Daten-Files und für die hier vorgestellte Frontend-Variante Geos ab 2.X. Ihr C 64 oder C 128 (nur 40-Zeichen-Mode) sollte mit einem oder mehreren Laufwerken ausgerüstet sein, die mindestens 331 KByte freien Speicherplatz zu Verfügung haben.

Als Speichermedien eignen sich alle unter Geos ansprechbaren Laufwerke - Diskstationen, REUs, Festplatten oder auch 64net-Laufwerke. Sollten Sie eine RAM-Erweiterung als Laufwerk einsetzen, dürfen Sie nicht vergessen, die Daten regelmäßig zu sichern.

Die Installation

Das Programm auf der Heft-Diskette ist voll lauffähig. Kopieren Sie sich die Software und die Datenfiles unter Geos auf das gewünschte Laufwerk und konfigurieren die einzelnen Laufwerke mit Hilfe des Programms "T&WSetup". Die Daten werden in einem cfg.-File gesichert.

Wenn Sie dann vom Spielleiter ein Ergebnisfile bekommen haben, können Sie mit dem Spiel beginnen. Zum Testen von "Trade & War" haben wir zwei Dateien mit Ergebnissen auf die Diskette zum Heft gespielt. Wählen Sie zum Probespielen als Spielerzahl "01" und als Rasse ebenfalls "01". Nachdem die Einstellungen durch das Programm getroffen wurden, gelangen Sie in das Spiel.

Das Spiel

Als Teilnehmer schlüpfen Sie in die Rolle eines interplanetaren Kommanders. Dabei können Sie sich wahlweise zu 15 verschiedenen galaktischen Rassen hingezogen fühlen.

Die Zugehörigkeit zu einer der 15 Gruppen, bestimmt die Strategie und die Möglichkeiten des Teilnehmers im Spielverlauf.



Mit dem Setup-Utility von "Trade & War" kann der Spieler das Programm an seine Hardware -Konfiguration anpassen

Im Spiel werden der Heimatplanet und die Raumschiffe gezeigt. Details werden durch einfachen Klick in die Sternenkarte auf den Bildschirm geholt. Die Gleiter, Fregatten und anderen Fahrzeuge können zur Erkundung ins All geschickt werden.

Kommt ein Schiff oder ein neuer Planet in Sichtweite, wird der Spieler erst in der nächste Runde darüber informiert. Es ist möglich, die eigene Raumflotte auszubauen und neue Gefährte zu produzieren. Grundlage dafür sind eine gut entwickelte Industrie und genügend Energie-Reserven. Neuentdeckte Planeten, werden durch die Spieler kolonialisiert und für die Entwicklung der eigenen Streitmacht genutzt.

Eine ausführliche Spiel-Beschreibung finden Sie im GeoWrite-Format auf der Diskette im Heft.

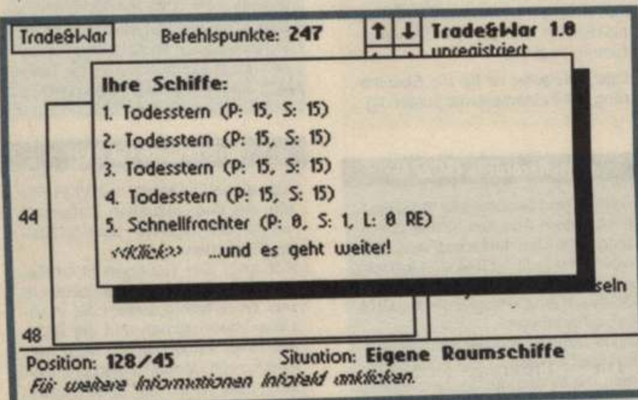
Die PC-Host-Software

Damit "Trade & War" gespielt werden kann, benötigt man einen Spielleiter mit einem PC. Hier wird die PC-Software gestartet, die die Datenfiles der Mitspieler auswertet und die neue Spielsituation berechnet. Der Einsatz eines PC ist aus Geschwindigkeitsgründen notwendig und durch die hohe Verbreitung als Mailbox-System vorteilhaft.

Das Programm für den PC ist Freeware und beim Autor erhältlich. Informationen zur Funktionsweise des Programms und zum Einsatz in einer Mailbox, finden Sie in der Anleitung auf der Diskette zum Heft bzw. beim Programm selbst.

Jörn-Erik Burkert

Info: Olaf Dzwiza, Stolzestr. 18, 30171 Hannover, FidoNet: OlafDzwiza 2:2437/40.24, GeoBoxNet: OlafDzwiza 230:233/0.0



Im Spiel werden alle Einstellungen für Schiffe, Planeten und Produktionsstätten getroffen



Die PC-Host-Software wertet alle Spielergebnisse aus und errechnet die neue Spielsituation

Geos-Systemroutinen

Geos intern

Folge 8

Die Systemdatei "Geos Kernel", Dreh- und Angelpunkt der beliebten C-64/C-128-Benutzeroberfläche, haben wir schon in ihre Bestandteile zerlegt – jede Menge phantastischer Assembler-Routinen entdeckten wir dabei. Heute stellen wir Ihnen Disketten-Routinen der Low-Level-Klasse vor, gehen auf die VLIR-Unterprogramme ein und beenden damit unsere Safari ins Innere von Geos.

Die restlichen Unterprogramme zur Disketten- und Laufwerksverwaltung unterteilen sich in zwei funktionelle Klassen:

Low-Level/Very-Low-Level-Routinen:

Sie manipulieren einzelne Blöcke auf Disk, allerdings keine zusammenhängenden Files.

VLIR:

Dieser Routinentyp befaßt sich ausschließlich mit der eigens vom Geos-Hersteller Berkeley Softworks entwickelten speziellen Art der Dateiverwaltung mit entsprechenden Laufwerken. Sie ähnelt entfernt der Manipulation von REL-Dateien im normalen DOS des C-64/C-128-Betriebssystems.

Die Geos-Druckertreiber holen ihre Parameter aus einer Standard-Einsprungtabelle, die für jeden Drucker die gleiche Funktion erfüllt. Acht Geos-Kernel-Programme sorgen für korrekte Druckausgabe – egal, ob Text oder Grafik.

Es bietet sich an, lädt man Druckertreiber per Geos-Routine GetFile. Bei Geos 64 muß er sich unbedingt auf derselben Disk wie das Druck-File selbst befinden, unter Geos 128 versteckt sich der Treiber im verdeckten Frnt-RAM ab Adresse $DF80. *bl*

ReadBlock ($C21A)

liest einen Sektor von der aktuellen Diskette in den Computerspeicher. Voraussetzung ist, daß die Routinen *EnterTurbo* und *InitForIO* bereits aufgerufen wurden. Um solche Speicherzugriffe wieder rückgängig zu machen, ist *DoneWithIO* zu aktivieren. Unser Listing zeigt, wie man *ReadBlock* in eigenen Programmen einsetzt..

Parameter:

R1: Track, Sektor des einzulesenden Blocks auf Diskette,
R4: Zeiger auf die RAM-Adresse, ab der man den Inhalt des Blocks ablegen will.
Nach Routinenaufruf steht die jeweilge Fehlernummer im x-Register ($00 = kein Fehler).

Demo-Listing zu ReadBlock ($C21A)

```
LiesBlock:    jsr EnterTurbo        ;Speed-Routine aktivieren
              txa                   ;x-Register auf Fehler
                                     prüfen
              bne Error             ;ja, dann zur Fehlerroutine
              jsr InitForIO         ;IO-Bereich einschalten
              ... weiterer Programm-Code ...
              LoadB r1L,#Spur       ;Track auf Disk
              LoadB r1H,#Block      ;Sektor
              LoadW r4,diskBlkBuf;Zeiger auf RAM
              jsr ReadBlock         ;Sektor-Inhalt lesen
              ... weiterer Programm-Code ...
              jmp DoneWithIO        ;IO-Bereich verlassen
Error:        rts                   ;zurück, wenn x-Reg. <>0
```

ReadLink ($904B)

identfiziert die ersten beiden Bytes eines Diskettenblocks (Track- und Sektornummern des folgenden Datenblocks). Mit dieser Routine läßt sich z.B. sehr schnell die Spuren- und Blockverkettung einer Datei auf Diskette nachvollziehen.
Achtung: diese Routine gibt´s nicht bei den Floppy-1541-Treibern (hier sollte man auf *GetBlock* ausweichen).

Parameter:

R1: Track-, Sektornummer des zu lesenden Blocks,
R4: Zeiger auf einen 256 Byte großen RAM-Puffer (z.B. diskBlkBuf), in dem die beiden relevanten Bytes (Spur, Sektornummer des Folgeblocks) abgelegt werden.
Das x-Register enthält die Fehlernummer.

PutBlock ($C1E7)

schreibt per Einsatz der Kernel-Routinen *InitForIO, EnterTurbo, WriteBlock* und *DoneWithIO* den Inhalt eines Diskettenblocks (254 Byte) auf die aktuelle Disk im Laufwerk.
Die Routine ist ähnlich konzipiert wie *GetBlock*: Es wird lediglich ein Sektor geschrieben. Sind´s aber mehrere, sollte man *WriteBlock* verwenden.

Parameter:

R1: Track- und Sektornummer auf Disk,
R4: RAM-Adresse der Daten im Speicher
Die Fehlernummer findet man im x-Register.

WriteBlock ($C220)

liest wie *PutBlock* Daten eines Sektors (254 Byte) aus dem Speicher und legt sie auf Diskette ab. Dabei wird vorausgesetzt, daß die Routinen *EnterTurbo* und *InitForIO* bereits aktiviert wurden. Außerdem sollte man das eigene Unterprogramm wieder mit dem Aufruf fvon *DoneWithIO* abschließen.

Parameter:

R1: Track und Sektornummer des Blocks, der die Daten enthalten soll,
R4: RAM-Adresse des Speicherbereichs, der die zu übertragenden Daten enthält (254 Bytes).
Das x-Register speichert wie gewohnt die Fehlernummer, nach Routinenaufruf werden die Inhalte von Akku und y-Register zerstört.

VerWriteBlock ($C223)

überprüft, ob die Übertragung eines Diskettenblocks per *PutBlock* bzw. *WriteBlock* korrekt abgelaufen ist. Falls nicht, beeilt sich die Routine, den Blockinhalt nochmals zu schreiben (per *WriteBlock*). Auch hier sind die Standard-Kernel-Routinen *EnterTurbo* sowie *InitForIO* unbedingt vorher zu aktivieren und per *DoneWithIO* zu schließen – außer, Sie wollen noch weitere Sektoren überprüfen lassen. Setzen Sie VerWriteBlock immer erst dann ein, wenn alle vorgesehenen Datenblöcke bereits auf Disk geschrieben wurden. Wie gewohnt, stehen in **R1** Track und Sektor, in **R4** der RAM-Bereich.

OpenRecordFile ($C277)

öffnet eine VLIR-Datei und bereitet alle Register zur weiteren Verwendung dieses Files mit anderen VLIR-spezifischen Routinen vor.

Falls das File noch keine Datensätze enthält, bekommt die Systemvariable curRecord den Wert 255 ($FF), ansonsten die Kennziffer für den ersten Datensatz (also $00).

Achtung: es darf stets nur eine VLIR-Datei geöffnet sein! Andere Diskettenroutinen kann man jedoch unbesorgt benutzen.

Parameter:

R0: Zeiger auf den Namen des zu öffnenden VLIR-Files ($00 als Endekennzeichen),

R5: weist auf die Adresse innerhalb diskBlkBuf, bei der der Directory-Eintrag beginnt.

Das x-Register ist für die Speicherung der Fehlernummer zuständig.

CloseRecordFile ($C277)

schließt und beendet alle VLIR-Routinen. Intern wird das Kernel-Unterprogramm UpdateRecordFile aufgerufen, die jede VLIR-Datei korrekt auf Diskette abschließt.
Dieses Kernel-Programm braucht keine Parameter.
Nach Beendigung der Routine enthält das x-Register die Fehlernummer; die Inhalte der Systemregister fileWritten und diskBlkBuf werden total verändert.

UpdateRecordFile ($C295)

überträgt sämtliche VLIR-Datei-Informationen inkl. geänderter BAM (Blockbelegungsplan) auf die aktuelle Disk im Laufwerk. Infos zum VLIR-File sind z.B. Indextabelle oder das aktuelle Datum.

Die Routine kann auf weitere Parameter verzichten.

Die Fehlernummer findet man im x-Register; der Inhalt der Systemvariablen *fileWritten* wird aktualisiert.

PointRecord ($C280)

ermöglicht dem Anwender, einen bestimmten Datensatz als aktuell gültigen in curReord in einem geöffneten VLIR-File einzutragen.

Der Datensatz läßt sich nun mit ReadRecord beliebig lesen oder mit WriteRecord überschreiben. Dazu ist im Akku die gewünschte neue Datensatznummer zu übergeben (denken Sie daran, daß der Computer die erste Datensatznummer stets mit "0" definiert).

Nach Aktivierung der Routine ist das x-Register für die Fehlernummer zuständig; im Akku und curRecord findet man die neue Datensatznummer, in R1 Track und Sektor des ersten Datenblocks des VL IR-Files.

NextRecord ($C27A)

Nach Aufruf dieser Routine wird der dem aktuellen folgende Datensatz als gerade gültiger in *curRecord* eingestuft. Parameter dürfen entfallen.

Das x-Register kümmert sich um die Fehlernummer; im Akku und R1 stehen nach Routinendurchlauf Track und Sektor des neuen Datensatzes.

InsertRecord ($C289)

fügt einen Datensatz ins VLIR-File ein. Ab dem aktuellen Datensatz werden alle folgenden um eine Stelle verschoben.
Achtung: Die Routinen InsertRecord, DeleteRecord, AppendRecord und WriteRecord ändern die VLIR-Datei-Informationen und die BAM, allerdings werden sie erst nach Aktivierung von UpdateRecord und CloseRecordFile auf der entsprechenden Diskette verewigt (sonst gelten die Änderungen nur im Speicher).

AppendRecord ($C289)

arbeitet prinzipiell wie InsertRecord, allerdings gibt's einen Unterschied: der neue Datensatz wird unmittelbar an den aktuellen (also nach dem in curRecord vermerkten) eingefügt. Alle dahinterliegenden Datensätze werden um eine Stelle verschoben (curRecord erhält nun die Nummer des jetzt aktuellen Datensatzes). Parameter und Registeränderungen sind mit InsertRecord identisch und haben dieselbe Funktion.

PreviousRecord ($C27D)

setzt den Inhalt von curRecord auf den Datensatz, der vor dem aktuellen liegt. Parameter sind nicht nötig. Funktion und Bedeutung des Akku, des x- und y-Registers sowie von R1 entsprechen den Kriterien von PointRecord.

DeleteRecord ($C283)

löscht den aktuellen Datensatz (er wird aus der Indextabelle entfernt). Alle folgenden Datensätze verschieben sich um eine Stelle nach vorn. Die BAM wird entsprechend angepaßt.
Beachten Sie, daß wie bei InsertRecord diese Daten lediglich im Speicher geändert werden – erst mit den entsprechenden Kernel-Routinen werden die geänderten Einträge auf Disk verewigt.
DeleteRecord braucht keine Parameter; nach Routinenaufruf steht die Fehlernummer wie üblich im x-Register.

ReadRecord ($C28C)

holt den aktuellen Datensatz von Diskette an einen frei wählbaren Platz im RAM. Zusätzlich läßt sich die Anzahl der zu lesenden Bytes begrenzen. Intern wird die Routine *ReadFile* aufgerufen.

Parameter:

R7: Zeiger auf die RAM-Adresse,wo die Daten abgelegt werden,
R2: maximale Anzahl der Bytes. Achtung: Sie darf 32258 Byte nicht überschreiten – das entspricht exakt dem Byte-Inhalt von 127 Blocks auf Diskette.
Nach dem Aufruf der Routine steht im x-Register die Fehlernummer. Der Akku registriert, ob der Datensatz leer ist ($00) oder Bytes enthält ($FF). In R7 findet man dann den Zeiger aufs zuletzt gelesene Daten-Byte. (ansonsten wird das Systemregister R7 nicht verändert). Findet man im x-Register die Fehlernummer $0B (BFR_OVERFLOW), erhält man in R1 die Track- und Sektornummer des Blocks, der die maximale Datenlänge überschritten hätte (wäre er vollständig geladen worden).

SetNLQ ($7915)

aktiviert den Schönschriftmodus. Diese Routine sollte man stets nach StartASCII aufrufen. Dabei wird in R1 für den Druckertreiber derselbe Arbeitsspeicher zur Verfügung gestellt. SetNLQ verändert nach dem Aufruf alle Register und die Inhalte von R0 bis R15.

WriteRecord ($C28F)

schreibt Daten in den aktuellen Record auf Diskette. Achtung: Diese Routine trägt lediglich die nackten Daten-Bytes ein – VLIR-File-Informationen und Blockbelegungsplan werden nur per Aufruf der Routinen *UpdateRecordFile* bzw. *CloseRecordFile* wieder auf Diskette übertragen. Ein eventuell vorhandener Datensatz wird sogar gelöscht!

Parameter:

R7: Zeiger auf die Adresse, ab der man die Daten im Computer-RAM findet,
R2: Anzahl der Bytes, die auf Disk übertragen werden sollen. Wie bei ReadRecord ist dabei peinlich darauf zu achten, daß es nicht mehr als 32258 Byte pro Record sind!
Das x-Register speichert die Fehlernummer; die Inhalte des Akku, y-Registers sowie von R0 bis R9 werden nach Routinendurchlauf zerstört. Der aktuelle Zustand der Systemvariablen *fileHeader, fileSize, fileTrScTab, curdirHead, dir2Head* (1571, 1581) und *fileWritten* verändert sich.

InitForPrint ($7900)

bereitet nicht den Drucker selbst, sondern den entsprechenden Treiber zur Druckausgabe vor: bei den meisten handelsüblichen 9-Nadel-Geräten besteht der Programm-Code dieser Routine lediglich aus *rts*. Bei älteren Druckern (z.B. MPS 801) mußte ein spezieller Datenpuffer angelegt werden, damit Geos auch diesen 7-Nadler unterstützt. Bei anderen Treiberprogrammen wiederum verwendet man InitForPrint zum Einstellen der Druckqualität.
Die Kernel-Routine darf die Inhalte aller Geos-Systemregister von R0 bis R15 verändern. Beim Entwurf eigener Druckertreiber sollte man das x-Register stets mit dem Wert $00 belegen – InitForPrint erzeugt nämlich intern automatisch eine Fehlermeldung. Die Programmierer des Geos-Systems empfehlen deshalb, in eigenen Programmen nach dem Aufruf von *InitForPrint* auf die Abfrage des x-Registers zu verzichten. GeoFile macht das z.B. nicht und erzeugt Fehlermeldungen, obwohl der Druckertreiber auf der Diskette ist. Nach dem Laden des entsprechenden Druckertreibers sollte man InitForPrint als allererste Routine aufrufen.

StartASCII ($7912)

bereitet den Drucker für Textausgabe vor und setzt den korrekten Zeilenabstand. Außerdem wird überprüft, ob der Printer online ist.
Dazu übergibt man in R1 einen 16-Bit-Wert als Zeiger auf den Arbeitsspeicher, den der Druckertreiber benutzen darf. Das x-Register gibt an, ob das Treiberprogramm betriebsbereit ist ($00) oder nicht. StartASCII verändert alle Register sowie die Inhalte von R0 bis R15.
Die Routine exisitiert im Geos-Kernel erst ab Version 1.3. Wer nicht sicher ist, ob der verwendete Treiber auf die Routine eingeht, sollte die Class abfragen: Alle Druckertreiber mit der Klassifizierung "Printerdriver 2.0" und höher (bei Farbdruckern: ClrPrntDrvr V2.1) sind für ASCII-Ausdrucke geeignet.

StartPrint ($7903)

macht den Printer für Grafikdruck bereit. Dazu ist in R1 ein Zeiger auf einen 1920 Byte großen RAM- Puffer zu setzen, den der Druckertreiber nutzen kann (aber nicht unbedingt muß). Im Prinzip stellt StartPrint nur den korrekten Zeilenvorschub für den Grafikausdruck ein. Außerdem testet die Routine, ob das Gerät betriebsbereit ist (dann steht der Wert $00 im x-Register). Nach dem Aufruf von StartPrint lassen sich alle Systemregister nutzen und mit beliebigen Werten beschreiben.

PrintBuffer ($7906)

druckt eine komplette Grafikzeile im Commdore-üblichen Hires-Format (8x8-Pixel-Struktur).Vom ersten bis zum achten Grafik-Byte wird wie gewohnt zunächst die erste Bit-Zeile gedruckt, darunter die zweite usw., bis zur achten – dann beginnt neben der ersten oben der Druck von Bit-Zeile 9. Standard sind 640 Bildpunkte pro Druckzeile. Ältere Commodore-Drucker (z.B. MPS 801) können jedoch nur 480 Pixel pro Druckzeile ausgeben – dann läßt *PrintBuffer* die überzähligen automatisch weg.

Parameter:

R0: Zeiger auf die zu druckenden Grafikdaten im Computer-Speicher. Eine Druckzeile besteht aus 8 x 80 Cards (= 640 Byte).
R1: Der in diesem Register gespeicherte 16-Bit-Wert weist auf einen 1920 Byte großen Datenpuffer, der auch bei *StartPrint* seine Bedeutung hat und beim Ausdruck der Grafikseite nicht verändert werden darf.
R2: Zeiger (WORD) auf die Farbinformation (80 Byte). Beim Parameter $0000 wird lediglich schwarzweiß gedruckt (der Wert in R2 ist daher nur bei Farbdruckern relevant).
Nach dem Aufruf der Routine muß man davon ausgehen, daß sich alle Werte in R0 bis R15 ändern (Geos-Programmierer sollten das berücksichtigen!). Ein automatischer Zeilenvorschub nach jeder Grafikzeile ist in der Routine integriert. Anschließend kann man in R0 bis R2 die Zeiger nach Wunsch auf die nächsten Druck-Informationen oder Grafik-Daten richten.

GetDimensions ($790C)

gibt Infos für den Grafikdruck aus: Breite (in Cards) und Länge (in Zeilen). Weitere Parameter benötigt die Routine nicht.
Nach dem Aufruf steht die Seitenbreite in Cards im x-Register, die Anzahl der maximal zu druckenden Zeilen ist im y-Register zu finden. *GetDimensions* ist lediglich beim Grafikdruck relevant.
Beispiel: Wenn man GeoWrite 2.1 startet, lädt das Programm zunächst den installierten Druckertreiber und ruft GetDimensions auf, damit sich die korrekte Seitenlänge des Dokuments an den entsprechenden Drucker anpaßt.

StopPrint ($7909)

unterbricht den Druckvorgang (Text oder Grafik) und löst einen Seitenvorschub (FormFeed) aus. Man kann die Routine entweder am Ende jeder Druckseite einsetzen oder dann, wenn der Anwender das Abbruchfeld in der Printer-Dialogbox anklickt.
Die Abbruchbedingung muß allerdings im Hauptprogramm definiert sein; der jeweilige Druckertreiber ist dafür nicht zuständig.

Parameter:

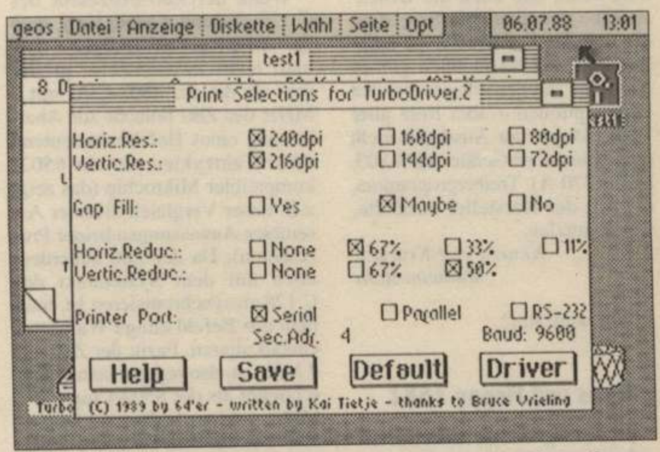
R0: wird mit $0000 belegt,
R1: Zeiger auf 1920 Byte großen Datenpuffer (s. *PrintBuffer*).
StopPrint verändert dieselben Register und Speicherinhalte wie *PrintBuffer.*

PrintASCII ($790F)

druckt beliebige ASCII-Zeichenketten aus, die mit $00 als Endekennzeichen abgeschlossen sein müssen. Erlaubt sind Zeichen mit den Codes 32 bis 126 und "Carriage-Return" (Code 13). Bei Commodore-Druckern werden die Bytes ins spezielle Commodore-Format konvertiert.

Parameter:

R0: Zeiger auf den RAM-Bereich, der die zu druckenden Zeichen enthält,
R1: weist auf den bei *StartASCII* definierten Arbeitsspeicher.
Ändert alle Systemregister.



Das Druck-Menü von "TurboDriver 2" greift wie alle anderen Geos-Drucker-Tools ausgiebig auf die Kernel-Routinen zu



Pin-Belegung des DFÜ-Kabels

Problem von Daniel Gutsche in der 64'er 8/95: Wir haben uns zur stationären Verbindung mit einem anderen C 64 ein DFÜ-Kabel gebastelt – allerdings vermuten wir, daß die Pin-Belegung nicht stimmt.

Zur Datenübertragung zwischen zwei C-64-Computern verwendet man am besten ein RS232-Kabel. Es sollte die maximale Länge von zwei Metern nicht überschreiten. Normalerweise reicht ein ungeschirmtes siebenpoliges Kabel (korrekte Verbindung der Pins s. anschließender Kasten).

Andreas Rathke, Kamenz

RS232-Kabel (Pin-Belegung)

Userport-Stecker 1	Userport-Stecker 2
A (Masse)	A (Masse)
B + C (RXD)	M (TXD)#
D (RTS)	K (CTS)
E (DTR)	L (DSR)
K (CTS)	D (RTS)
L (DSR)	E (DTR)
M (TXD)	B + C (RXD)

Druckfunktion von Chess-Card

Vor kurzem habe ich mir die spielstarke "Final Chess-Card" zugelegt. In der Bedienungsanleitung steht, daß auch die Druckausgabe des aktuellen Spielstandes und der einzelnen Züge möglich sein soll. Tatsächlich gibt es ein Menü mit den entsprechenden Druckoptionen – aber trotz aller Tricks klappt der Ausdruck nicht (angeschlossene Geräte: MPS 803, MPS 1270-A). Treiberprogramme, die mir der Hersteller zusandte, waren nutzlos.

Klaus-Peter Krücker, Mülheim-Ruhr

Wer weiß Rat?

Exos und Floppy 1581

Seit einiger Zeit arbeite ich auf meinem C 64 mit dem Betriebssystem "Exos" (veröffentlicht im 64'er-Sonderheft 84). Allerdings habe ich mir jetzt die Diskettenstation 1581 zugelegt – doch damit klappt der Schnellader überhaupt nicht. Nach der Meldung "Loading" stürzt der Computer ab, das Laufwerk bleibt stehen (obwohl die LED leuchtet). Zur eigentlichen Datenübertragung in den Speicher kommt es erst gar nicht. Weiß jemand, wie man Exos ans Betriebssystem der Floppy 1581 (DOS 10.0) anpaßt?

Daniel Krug, Mörlenbach

Exos wurde seinerzeit für die DOS-Version 2.6 der Floppy 1541 entwickelt – um die Systemerweiterung für die 1581 kompatibel zu machen, sind tiefgreifende Programmänderungen nötig. Das ist bis jetzt noch keinem gelungen.

Red. 64'er

Z80-Prozessor mit 4 MHz

Zum Thema "Taktfrequenz des Z80 im C 128" (erörtert in der Beantwortung der Leserfrage von Carsten Böhle in der 64'er 7/95) möchte ich bemerken: Der reine Wert der Taktfrequenz erlaubt noch lange keine Beurteilung darüber, ob ein Mikroprozessor schneller ist als der andere! Der 68040 von Motorola mit 25 MHz ist z.B. schneller als der i860 von Intel mit 40 MHz (wurde in diversen Praxistests festgestellt).

Wenn der Z80-Prozessor des C 128 also mit 4 MHz läuft, muß die Arbeitsgeschwindigkeit deshalb nicht zwangsweise größer sein als die der 8502-CPU mit 2 MHz: der Z80 braucht zur Abarbeitung eines Befehls bedeutend mehr Taktzyklen als ein 6502-kompatibler Mikrochip (das zeigt z.B. unser Vergleich diverser Assembler-Anweisungen beider Prozessoren). Da der Z80 außerdem noch mit dem Systemtakt des C 128 zu synchronisieren ist, muß man pro Befehl einige Waitstates einkalkulieren. Fazit: der Z80 mit 4 MHz ist also auf gar keinen Fall schneller als der 8502-Chip mit 2 MHz, sondern bedeutend langsamer! Die auf dem Rockwell-6502-RISC-Prozessor basierenden CPUs sind im Vergleich mit dem auf dem Intel-8080 basierenden Zilog Z80-CISC-Prozessor erheblich schneller. Hat man aber vor, komplexe Programme zu generieren, sind solche Prozessoren in Assembler nahezu unprogrammierbar.

Thomas Bodlien, Hillerse

Will ich z.B. das Directory laden, meldet der Computer "Searching for $" – sonst passiert nichts. Welche ICs der 1571 oder des C 64 muß ich auswechseln, damit die Floppystation wieder einwandfrei funktioniert?

Lars Wienert, Suterode

Taktzyklen-Vergleich Z80 – 8502 (6510/8500/65816)

Z80		8502(6510/8500/65816)	
Befehl	Takte	Befehl	Takte
ld a,00	7	lda #0	2
adc a,01	7	adc #1	2
ld (hl),xxxx			
inc (hl)	7+11=18	inc $xxxx	6
neg	8	eor #$ff	2
call xxxx	17	jsr $xxxx	6
ret	10	rts	6

DIP-Schalterproblem

Mein C 128 ist am Epson-Drucker FX-1050 angeschlossen (per 1,24-m-langem Parallelkabel am Userport, verbunden mit der Centronics-Schnittstelle des Druckers). Leider klappt es nicht, unter Geos längere Texte oder Grafiken auszudrucken. Die Stellung der DIP-Schalter: 1-1: on, 1-2: off, 1-3: on, 1-4: off, 1-5: on, 1-6: on, 1-7: off, 1-8: on, 2-1: off, 2-2: off, 2-3: off, 2-4: on.

Wer weiß Rat bzw. kennt den richtigen Druckertreiber und kann mir nähere Infos zum Centronics-Parallel-Interface MP 201 mit 9 Volt Versorgungsspannung geben?

Peter Weiß, Bremen

Floppy läuft und läuft ...

Problem von MatthiasKorn in der 64'er 7/95: Nach dem Einschalten beginnt meine Floppy zu laufen und hört nicht mehr auf damit – obwohl gar keine Disk im Laufwerk liegt.

Vor einigen Jahren hatte ich dasselbe Problem mit meiner 1541-II.

Zunächst glaubte ich, das serielle Kabel sei schuld daran, aber als nach dem Umbau meiner Anlage plötzlich auch die Zweitfloppy streikte, war's klar: mein Netzteil hatte sich verabschiedet. Und deshalb bekam das Laufwerk nicht mehr genügend Strom, um den Einschalttest zu beenden.

Neue Netzteile gibt's nach wie vor beim bekannten Fachhandel.

Josef Dean Butler, Mannheim

Ich habe den C-64-Brotkasten und eine1571 als Diskettenstation. Versehentlich wurde der Netzstecker aus der Buchse gezogen während das Laufwerk noch lief – und jetzt bleibt es nach jedem Einschalten nie wieder stehen.

RAM-Erweiterung 1764

Langsam wird mir die Kapazität dieses RAM-Moduls (256 KByte) im Betrieb mit meinem C 128 zu knapp. In einer früheren Ausgabe der 64'er stand, daß man die 1764 mit einem simplem Trick in eine REU 1750 verwandeln kann: per Manipulation des Jumpers unter dem Prozessor-Chip der RAM-Erweiterung.

Allerdings besitzt die REU auch noch zusätzliche Steckplätze für weitere RAM-Bausteine. Reicht es also, lediglich den Jumper zu durchtrennen, um dann 512 KByte Speicherplatz in der RAM-Erweiterung nutzen zu können, oder sind noch zusätzliche RAM-Chips nötig, die man in den entsprechenden Steckplätzen unterbringt?

Roland Krautt, Karlsruhe

Programme nachladen

Ich suche schon seit langem nach einem Trick, innerhalb eines Basic-Programms ein anderes nachzuladen. Folgende Zeilen führten zumindest zum Teilerfolg:

```
10 A$="(Name)"
20 SYS 57821 A$,8,1
30 POKE 780,0: SYS 65493
```

Das zweite Programm, dessen Dateiname in der Variablen A$ gespeichert ist (Zeile 10), wird zwar problemlos geladen – aber dann bricht der C 64 mit der Fehlermeldung "SYNTAX ERROR IN 30" ab. Was mache ich falsch? Liegt es am Aufruf der Systemroutine?

Frank Hornemanns, Köln

Hinweis: Sowie Leser uns Problemlösungen zusenden, werden diese individuell an den Fragesteller weitergeleitet. Die Veröffentlichung zu Gunsten aller Leser folgt im nächst erreichbaren Heft. *Die Red.*

Highlights

Programm-Service-Disk

64'er 9/95

Diskette Seite A
Tool-Station 5.1 AssBlaster V3.1 Packer-Software: Hackpack 4.0 Tips & Tricks zum C 64: u.a. Pulldown-Demo, Dir-Basic, Format-Print

Diskette Seite B
Brandneues DFÜ-Spiel unter Geos (Demo-Version): Trade & War



Kleinanzeigen-Auftrag für den COMPUTER-MARKT

Bitte veröffentlichen Sie in der nächsterreichbaren Ausgabe von 64'er den folgenden Kleinanzeigentext unter der Rubrik

_______________ (z.B. C 64, C 128, Plus/4, Software, Zubehör, Verschiedenes).

Meine Anzeige ist eine ☐ **private Kleinanzeige** (5 Zeilen mit je 32 Buchstaben, maximal 160 Zeichen)

☐ DM 5,– liegen ☐ bar ☐ als Scheck bei. **Bitte keine Briefmarken!**

Bezahlung über Postscheckkonto nicht mehr möglich.

Meine Anzeige ist eine ☐ **gewerbliche Kleinanzeige für DM 12,–** (zzgl.MwSt.) **je Druckzeile**

Anschrift:
64'er
Computermarkt
MagnaMedia Verlag AG
Postfach 1304
85531 Haar

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze

Absender:

Name/Vorname

Straße

PLZ/Ort

Telefon

Datum: Unterschrift:



Impressum

Chefredakteur: Harald Beiler (bl), verantwortlich für den redaktionellen Teil
Textchef: Jens Maasberg (jm)
Redaktion: Jörn-Erik Burkert (lb)
Redaktionelle Mitarbeit: Peter Klein, Denis Döhler, W. Weicht
Redaktionsassistenz: Sylvia Simon (sd)

So erreichen Sie die Redaktion:
Tel. 089/4613-202, Fax 089/4613-433, Btx *64064#
Hotline 089/4613-640, jd.. Mi., 15-17.00 Uhr

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten worden sein, so muß das angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in den von der MagnaMedia Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträgern. Mit Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von MagnaMedia Verlag AG verlegten Publikationen und dazu, daß die MagnaMedia Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Layout: Uschi Böcker, Dagmar Berninger
DTP-Operator: Dorothea Voss, Hans-Dieter Schimank
Titellayout: Uschi Böcker
Computergrafik: Alexander Gerhardt
Fotografie: Roland Müller

Anzeigenverkaufsleiter: Regine Schmidt (828), verantwortlich für den Anzeigenteil

So erreichen Sie die Anzeigenabteilung
Tel. 089/4613-962, Telefax 089/4613-394

Anzeigenverwaltung / Disposition: Susanne Schröder (853)
Anzeigenpreise: Es gilt die Preisliste Nr. 3 vom 01. 01. 1995

International Account Manager: Kurt Skupin (352)
Assistenz: Michelle Berner (360), Fax 089/4613-775

Anzeigen-Auslandsvertretungen:
Großbritannien: Smyth International, London, Tel. 0044-183140-5058, Fax 0044-181341-9602
Frankreich: Ad Presse International S.a.r.l. 34, rue de Camille Pelletan, F-92300 Levallois-Perret, Tel. 0033-1-47317530, Fax 0033-1-47317507
Taiwan: Acer TWP Co., Taipei, Tel. 008862-713-6959, Fax 008862-715-1950
Italien: Medias International, Mariano, Tel. 0039-31-751494, Fax 0039-31-751482
Holland: Insight Media, Laren, Tel. 0031-2153-12042, Fax 0031-2153-10572
Israel: Baruch Schaefer, Holon, Tel. 00972-3-556-2256, Fax 00972-3-556-6944
Korea: Young Media Inc, Seoul, Tel. 00822-765-4819, Fax 00822-757-5789
Hongkong: The Third Wave (H.K.) Ltd., Tel 00952-7640989, Fax 00852-7643857

Abonnement-Service:
64'er Aboservice
74168 Neckarsulm, Tel.: 07132/959-242,
Fax: 07132/959-244
Einzelheft: DM 9,80
Jahresabonnement Inland
(12 Ausgaben): DM 105,–
(inkl. MwSt., Versand und Zustellgebühr)
Jahresabonnement Ausland: DM 129,–
(Luftpost auf Anfrage)
Österreich: DSB-Aboservice GmbH,
Arenbergstr. 33, A-5020 Salzburg, Tel.: 0662/643866,
Jahresabonnementpreis: öS 864
Schweiz: Aboverwaltungs AG, Sägestr. 14,
CH-5600 Lenzburg, Tel.: 064/519131,
Jahresabonnementpreis: sfr. 105.–
Nachbestellung Einzelhefte:
64'er Leserservice
Heiner-Fleischmann-Str. 2, 74172 Neckarsulm
Tel. 07132/969-181 – Fax 07132/969-190
(Heftpreis + DM 6,– Versandpauschale)

Erscheinungsweise: monatlich (zwölf Ausgaben im Jahr)

Leitung Herstellung: Klaus Buck (180)

Technik: Sycom Druckvorstufen GmbH, Hans-Pinsel-Straße 2, 85540 Haar

Druck: Druckerei E. Schwend GmbH & Co. KG, Schmollerstraße 31, 74523 Schwäbisch Hall

Vertriebsleitung: Benno Gaab

Sonderdruck-Dienst: Alle in dieser Ausgabe erschienenen Beiträge können für Werbezwecke in Form von Sonderdrucken hergestellt werden.
Anfragen an:
Klaus Buck,
Tel. 089/4613-180,
Telefax: 089/4613-232

1995 MagnaMedia Verlag Aktiengesell

Vorstand: Carl-Franz von Quadt (Vors.), Kr Clifford, Eduard Unzeitig

Verlagsdirektor: Wolfram Höfler

Anschrift des Verlages:
MagnaMedia Verlag Aktiengesellschaft,
Postfach 1304, 85531 Haar bei München,
Telefon 089/4613-0, Telex 522052, Telefax 089/4613-100

Diese Zeitschrift ist auf chlorfreiem Papier mit einem Altpapieranteil von 100% gedruckt.Die Druckfarben sind schwermetallfrei.

64'er COMPUTER-MARKT

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »64'er« bietet allen Computerfans die Gelegenheit, für nur 5,- DM eine private Kleinanzeige mit bis zu 4 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **November-Ausgabe** (erscheint am 27.10.95): Schicken Sie Ihren Anzeigentext bis 21. September (Eingangsdatum beim Verlag) an »64'er«. Später eingehende Aufträge werden in der **Dezember-Ausgabe** (erscheint am 24.11.95) veröffentlicht.

Am besten verwenden Sie dazu den vorbereiteten Coupon im Heft.

Bitte beachten Sie: Ihr Anzeigentext darf maximal 4 Zeilen mit je 40 Buchstaben betragen.

Schicken Sie uns DM 5,- als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 12,- je Zeile Text veröffentlicht.

Private Kleinanzeigen

COMMODORE 64

Verk. C 64 II, 1541 II, FC 3, Joy LC 20, W&T-Interf. üb. 60 Disks, 60 64'er-Hefte, 3 Bücher, alle OK, zus. 400 DM; H. Sinapius, A.-Sandrock-Str. 101, 12627 Berlin

C 64 II + 2 x 1541 + Final Cartridge III + Software + Monitor; Preis: VHS; Tel. 0371/224828 nach 17 Uhr

Verk. C 64 II in PC-Gehäuse, 1451 II GeoRAM, CMD-PD 2000, VC 1531, Maus, 4 Joys, Geos 2.5, GeosLQ, Gateway, 46 64'er + 16 Sonderhefte, ca. 100 Disks VB 1000,–. Tel. 030/8253382, Jens

Verk. Netzteil C 64, FC III, MasterBase, GeoBasic, C 64'er-Spielesammlung 1 u. 2, Mega Assembler f. Geos, div. Geos-Orig.-Software. Adolf Köhler, Oschatzer Ring 22, 12627 Berlin, Tel. 030/9937452

Verkaufe Floppy 1541 + 1571 100% OK, je 80 DM. J. Brederech, Alemannenstr. 14, 76532 Baden-Baden

C 64 + 1541 + Jiffy-DOS eingeb. 195 DM. C 64 + 1541 + SpeedDOS + eingeb. 210 DM. C 64 + 1541 + Dolphin-DOS eingeb. 260 DM. 1541 I mit P.-Kab., 1541 II je 100 DM. Expansions. Wei 4 oder 5 fach je 50 DM. Tel. 02303/80916

Verk. C 64 II, 1541 II, Page-Fox, Mon 1802, Drucker LC10, Maus, Joystick, Spiele und versch. Programme ca. 80 Disks. Kpl. 700,– DM. Tel. 00436277/7109

Private Kleinanzeigen

COMMODORE 128

C 128 + 1571, oder C 128D + Jiffy-DOS eingeb. 295 DM. Fl. 1581 + Jiffy-DOS eingeb. 250 DM. BTX-Modul II 40 DM. Maus 1531 40 DM. Und vieles mehr. Tel. 02303/80916

C 128-Blech, FD 1581, REU 1750, Mon-1084, div. Zubehör 700,– DM VB. Tel. 05085/7016

Suche Farbmonitor mit Kabel für C 128 preiswert zu kaufen. Zuschr. an Scolasti, Mozartstr. 9, 01809 Heidenau

SOFTWARE

Suche orig. Spiel Maniac Mansion. Zahle bis zu 30 DM für 100% lauffähiges Spiel mit Anleitung. Adr.: Bert Töpelt, Wolkowstr. 8, 15517 Fürstenwalde. Tel. 03361/340719

Geos 64 2.0 40 DM, Deskpack 25 DM, Geochart 30 DM, Geofile 30 DM, Geocalc 30 DM. TopDesk 10 DM. M.-Pack1 20 DM. Vizarstar 64 40 DM. Und vieles mehr. Tel. 02303/80916

Suche, Multifont-Software Master-Diskette für Zeichensatz-Design der Commdore-Drucker MPS802/1526 von Ingenier-Büro Alfred Hollmann in Hamburg. Tel. 0041/1/8251204

Private Kleinanzeigen

ZUBEHÖR

Digital Genlock C 64/128 kompl. mit Videoprofi-Steckmodul, wenig gebraucht. Neupr. 998,– DM für 550,– DM VB. Videofox einschl. Kollektion Farbbilder und Vorspännen, Movies für den Videofox zus. 65,– DM. Tel. 05031/75217 oder tagsüber 0511/3622325

Verk. 64'er Einsteigerkurs m. Disk f. 20,– DM; 300-er PRG Softwarepaket + 1 Disk Textsystem 64 für 35,– DM, alles mit dt. Anleitung; Porto wird übernommen. Tel. 03683/600416

Verk. Das kompl. Schachprg. für 25,– DM; 64'er Sonderhefte je 8,– DM: SH 7/86 PEEKs + POKEs, SH 43 Tips, SH 47 Drucker, SH 42 + 66 Spiele, SH 53 Das Beste aus 5 Jahren. Tel. 03683/600416

Verkaufe Hardcopy-Modul für 30,– DM (NP: 50,– DM); Buch: Basic in 100 BSP. für 12,– DM; zus. nur 40,– DM Porto wird übernommen!!! Tel. 03683/600416 (nach 16 Uhr)

Verk.: Datasette für C 64/128; 64'er Hefte, Sonderhefte (auch C 128), teils mit Disk; Bücher, Software (C 64/128); Liste bei: Reiner Kurz, Jakob-Huber-Str. 12, 82110 Germering

Flash 8, 1 MB + FloppyDOS 350,–, Mon. 1802 250,–, Mouse 1351 25,–, 64'er Hefte 1/93-7/95 120,–, Bookw. Master Base + All. ü. Masch.-Spr. je 20,–, Userp. Exp. 3f. 10,–. Tel. 02372/60283

Private Kleinanzeigen

Wichtige Hinweise für alle Kleinanzeigen-inserenten

- Kleinanzeigenaufträge **ohne Absenderangabe** sowie Anzeigentexte unter **Postlagernummer** können leider **nicht** veröffentlicht werden.
- Bitte verwenden Sie für Ihren Auftrag das Formular auf Seite 20.
- Zur Bezahlung von Kleinanzeigen werden **keine Fremdwährungen** mehr angenommen.
- Bitte achten Sie auch darauf, daß Ihr Auftrag immer **vollständig ausgefüllt** ist (z.B. Unterschrift).

Reu 1764/512K 175 DM. 256K Eprompl. 50 DM. Pl. mit 2 S. 10 DM Brenner ab 40 DM. Löschg. 50 DM. Dolphin-D., Prologic-D. Pro-Speed 128 je 100 DM. Jiffy-D. 128 60 DM. IEEE Modul mit Soft 40 DM. Tel. 02303/80916

Verk. Epson SQ 870 Tintenstr. für 700,–. Sehr wenig gebraucht und 100% i.O., verarb. auch Endlospapier. Zuschr. bitte an: Thomas Stauch, Quergasse 5, 99819 Eckardtshausen

VERSCHIEDENES

Verk. StarTexter u. StarDatei für C 64 zus. 110,– DM sowie Mega Assembler für Geos 80,– DM, alles Originale, Vorkasse, zuzügl. 6,– DM Porto. W. Müller 05381/5544

Verk. C 64 II, 1541 (def.), Drucker Citizen, 3 Joysticks, Software (u.a. Geos 2.5) Preis VB evtl. Einzeln. Tel. 05941/6340 Neuenhaus

Verk. Wörterbuch Englisch 10000 Vokabeln 10,– 64'er Sports-Pack 20 Games 25,– C 64 Multi-Dat 25,– C 64 Manager 20,– Videoverwalt. 10,–. R. Schulz, Körtestr. 31, 10967 Berlin

Suche Floppy 1571 für C 128, kann auch leicht defekt sein. Telefon & Fax: 0561/69729, Citiruf: 01681/1155142

Gewerbl.Kleinanzeigen

Software, Telespiele u. Zubehör
Preisliste, Tel. 06447/285

C64/C 64 II/128 ab 80,–/100,–/120,–/Floppy 1541/1541II/1570/1571/1581 ab 80,–/100,–/120,–/130,–/200,– Monitore ab 80,– Drucker ab 80,– C 128D ab 200,–; versch. Soft- Hardware Zubehör uvm. Liste anfordern; solange Vorrat reicht, Tel. 0871/63829 oder Fax 0871/64995

Achtung:

Wir machen unsere Inserenten darauf aufmerksam, daß das **Angebot**, der **Verkauf** oder die **Verbreitung** von urheberrechtlich geschützter Software nur für Originalprogramme erlaubt ist.

Das Herstellen, Anbieten, Verkaufen und Verbreiten von **»Raubkopien«** verstößt gegen das Urheberrechtsgesetz und kann straf- und zivilrechtlich verfolgt werden. Bei Verstößen muß mit Anwalts- und Gerichtskosten von über DM 1000,- gerechnet werden.

Originalprogramme sind am Copyright-Hinweis und am Originalaufkleber des Datenträgers (Diskette oder Kassette) zu erkennen und normalerweise originalverpackt. Mit dem Kauf von Raubkopien erwirbt der Käufer auch kein Nutzungsrecht und geht das Risiko einer jederzeitigen Beschlagnahmung ein.

Wir bitten unsere Leser in deren eigenem Interesse, Raubkopien von Original-Software weder anzubieten, zu verkaufen noch zu verbreiten. Erziehungsberechtigte haften für ihre Kinder.

Der Verlag wird in Zukunft keine Anzeigen mehr veröffentlichen, die darauf schließen lassen, daß Raubkopien angeboten werden.

Erste Schritte mit GeoProgrammer



Geos zum Anfassen

Das einst beste Entwicklungspaket für Geos-Applikationen (MegaAssembler) ist vom Markt – doch GeoProgrammer schließt die Lücke. In dieser Folge unseres Programmierkurses zeigen wir Ihnen, wie man im VLIR-File auf Disk abgelegte Datensätze wieder lädt und auf den Bildschirm bringt.

Bevor wir uns in den nächsten Kursteil stürzen, müssen zwei Zeilen des Listings in der 64'er 8/95 geändert werden – im Label "Put-Routine":

```
LoadW r7, HinzuPuffer
LoadW r2,430
```

Der Grund: der Name der Karteikarte (also das Stichwort) würde sonst beim Laden niemals wieder auf dem Bildschirm auftauchen – inkl. Kartenname sind also zehn Eingabezeilen (= 430 Byte) zu berücksichtigen, die im Datensatz (Record) auf Disk abgelegt werden.

Das Speichern selbst ist kinderleicht: auch dafür hält das Geos-Kernel raffinierte Routinen bereit. Ändern Sie die entsprechenden Sprungmarken im Quelltext lt. unserem Listing "Datei aktualisieren und speichern".

Datensätze laden

Mit Unterstützung der komfortablen Geos-Kernel-Routinen lassen sich selbstverständlich auch die entsprechenden Unterprogramme zum Laden und für die Bildschirmausgabe ebenso problemlos realisieren.

Dazu müssen wir wieder unseren bisher erarbeiteten Quelltext ergänzen (am besten verwenden Sie das GeoWrite-File "CardBox.KURS" von der Programmservice-Diskette in der 64'er 8/95): Der Programmteil zum Laden gewünschter Datensätze (repräsentiert per Option "Öffen" im Menü "Datei" bzw. durchs Label "AcOeffnen" im Source-Code) ist zu ändern. Werfen Sie die Dummy-Anweisung "jsr ReDoMenu" raus und tippen Sie den neuen Quelltext (s. Listing S 23) dazu.

Der neue Programm-Code zieht einen Rattenschwanz weiterer Sprungmarken (Labels) nach sich: Zunächst erzeugt die Routine eine Eingabe-Dialogbox, in die Sie den gewünschten Dateinamen eintragen müssen. Per Geos-Kernel-Unterprogramm "FindFTypes" wird dann nach dem registrierten File-Typ gesucht (hier: Identifikation per Geos-Class). Der Quelltext zu den Labels **Laden** und **Lesen** ist dafür zuständig, daß der Inhalt des Datensatzes ins Computer-RAM geholt wird und sich in der Karteikartenfläche auf dem Bildschirm zeigt. Das unterstützen die beiden Systemroutinen "OpenRecordFile und MoveData". Dazu braucht man die Systemvariable fileHeader; die beiden nächsten Bytes hinter der darin angegebenen Position identifizieren Spur und Sektor des Beginn unserer VL IR-Datei auf Disk.

Dann ist der Record zu öffnen ("OpenRecordFile") und die aktuelle Datensatznummer ans Programm zu übergeben (Inhalt der Variablen "Record").

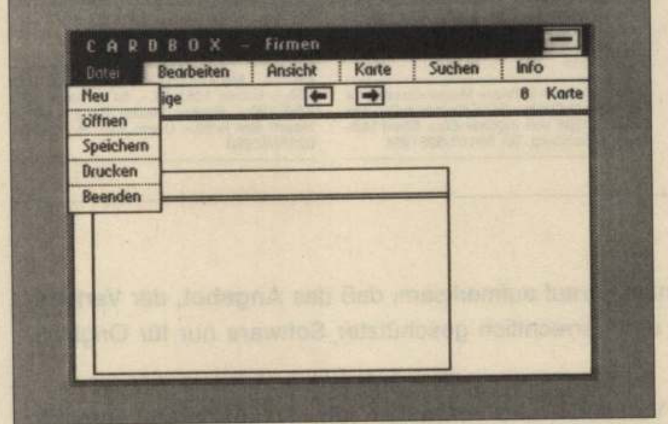
Um die Daten dann auf den Screen zu bringen, ist der Lesezeiger auf den gewünschten Datensatz zu positionieren ("PointRecord") und die Menge der auszugebenden Bytes zu bestimmen (zehn Zeilen mit jeweils 43 Zeichen, also inkl. Stichwort). Diese Funktionen erledigt der Programm-Code beim Label "Ausgabe". Vorher wird auch nach dem Lesen das File sofort wieder geschlossen ("CloseRecordFile").

Damit befindet sich der Datensatzinhalt zwar im RAM – auf dem Bildschirm läßt sich aber noch kein einziges Byte blicken. Dafür müssen Sie schon selbst sorgen: per Quelltext im Label "OutPut" (s. Listing). Sie können ihn unmittelbar hinter dem Label "Ausgabe" ins GeoWrite-Quelltext-File anfügen.

Es ist nicht zu vermeiden, daß der Programm-Code ebenso umfangreich ist wie der für die Dateneingabe: pro Ausgabezeile sind die Koordinaten zu bestimmen, an denen der Text aus dem RAM geholt wird und anschließend auf der Karteikarte erscheint. Außerdem ist pro Datenzeile die Kernel-Routine "Put-String" zu aktivieren (als Pendant zu "GetString").

Geben Sie jetzt den Text inkl. Stichwort für eine beliebige Karteikarte ein (Option "Neu" im Menü "Datei" und "Hinzufügen ..." im Menü "Karte"). Anschließend können Sie die Eingabe entweder per Schließ-Icon rechts oben beenden oder mit dem Menüpunkt "Beenden" in "Datei". Erst jetzt wird der soeben eingetragene Datensatz auf Disk verewigt – unter Berücksichtigung aller für die Floppy wichtigen Infos wie z.B. Aktualisierung der BAM usw.

Holen Sie probehalber diesen Datensatz auf den Bildschirm zurück: im Menü "Datei" die Option "Öffnen" anklicken und nach Aufforderung durchs Programm den entsprechenden Dateinamen eingeben. Wenn Sie die <RETURN>-Taste drücken, erscheinen automatisch die vorher eingegebenen Bytes auf dem Screen.

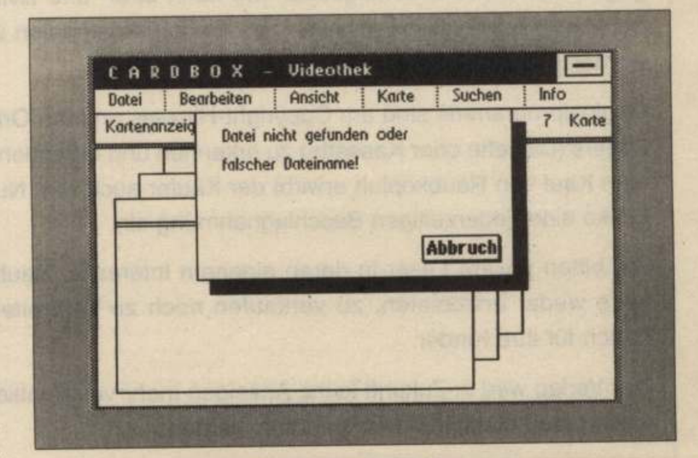


Die Menüpunkte "Öffnen" und "Beenden" – unser heutiges Thema

Integrierte Fehlerabfrage

Im Label "AcOeffen" ist die eine Routine eingebaut, die eine Fehler-Dialogbox erzeugt (Label "ErrBox"). Sie ist so allgemein gehalten, daß man sie quasi in jedes andere Geos-Programm übernehmen kann (es werden außer "ProgStart" keine weiteren internen Sprungmarken tangiert). Wie gewohnt wird das Fehlermeldungs-Window durch die Systemroutine "DoDlgBox" erzeugt – wichtig ist allein die exakte Definition der Parameter (s. Listing). Dazu muß man im Systemregister R0 einen 16-Bit-Wert als Zeiger auf die Parametertabelle übergeben. Die Labels "Err1" und "Err2" speichern den Text, der in der Dialogbox erscheinen soll.

Aus technischen Gründen war es nicht möglich, die Quelltexterergänzungen auf der Programmservice-Disk in diesem Heft unterzubringen. In der nächsten 64'er finden Sie jedoch das aktualisierte GeoWrite-File "CardBox.KURS" – auch mit den Änderungen der nächsten Kursfolge *bl*



Fehlermeldung per Kernel-Routine DoDlgBox

Label "AcOeffnen"

```
AcOeffnen:
          jsr DoPreviousMenu
          jsr DelFName ;alten Dateinamen löschen
          LoadW a0,FileName
          LoadW r0,FNmBox
          jsr DoDlgBox ;gewünschten Dateinamen eingeben
          jsr DelTitle
          jsr DelCdField
          jsr Dateiname ;eingetragenen Namen aus RAM holen
          LoadW r6,FileName
          LoadB r7L,3
          LoadB r7H,1
          LoadW r10,Class ;Identifikation nach Class-Name
          jsr FindFTypes
          txa ;Inhalt von x-Register in Akku
          beq Laden ;wenn "0", dann laden
          jmp ErrBox ;sonst Fehlermeldung ausgeben

Laden:    LoadW r0,FileName
          jsr OpenRecordFile ;Datensatz öffnen
          txa
          beq Lesen
          jmp ErrBox

Lesen:    jsr i_MoveData
          word fileHeader+2
          word Index
          word 2
          jsr Ausgabe
          jmp ReDoMenu
Index:    block 2 ;zwei Bytes mit 0 belegen

ErrBox:   LoadW r0,ErrorDB
          jsr DoDlgBox
          jmp ProgStart
ErrorDB:  byte $81 ;Fehlermeldung als Dialogbox ausgeben
          byte DBTXTSTR,16,16
          word Err1
          byte DBTXTSTR,16,32
          word Err2
          byte CANCEL,17,72 ;Dialog-Icon bestimmen
          byte NULL
Err1:     byte "Datei nicht gefunden oder",NULL
Err2:     byte "falscher Dateiname!",NULL

Ausgabe:  LoadW r0,FileName
          jsr OpenRecordFile
          lda Record
          jsr PointRecord
          LoadW r7,HinzuPuffer
          LoadW r2,(10*43) ;zehn Zeilen zu je 43 Zeichen
          jsr ReadRecord
          jsr CloseRecordFile
          jsr OutPut
          rts
```

© 64'er

Label "OutPut" (Daten auf Screen zeigen)

```
OutPut: LoadW r0,HinzuPuffer
        LoadB r1H,90 ;vertikale Position bei Ausgabe
        LoadW r11,20 ;horizonzale Position (erste Ausgabespalte)
        jsr PutString ;Geos-Kernel-Routine
        LoadW r0,FirstRow ;Eingabezeile 1
        LoadB r1H,105 ;y-Position
        LoadW r11,20 ;x-Position
        jsr PutString
        LoadW r0,SecRow ;Ausgabezeile 2
        LoadB r1H,114 ;y-Position
        LoadW r11,20 ;x-Position
        jsr PutString
        LoadW r0,ThirdRow ;Ausgabezeile 3
        LoadB r1H,123 ;y-Position
        LoadW r11,20 ;x-Position
        jsr PutString
        LoadW r0,ForthRow ;Ausgabezeile 4
        LoadB r1H,132 ;y-Position
        LoadW r11,20 ;x-Position
        jsr PutString
        LoadW r0,FifthRow ;Ausgabezeile 5
        LoadB r1H,141 ;y-Position
        LoadW r11,20 ;x-Position
        jsr PutString
        LoadW r0,SixthRow ;Ausgabezeile 6
        LoadB r1H,150 ;y-Position
        LoadW r11,20 ;x-Position
        jsr PutString
        LoadW r0,SevenRow ;Ausgabezeile 7
        LoadB r1H,159 ;y-Position
        LoadW r11,20 ;x-Position
        jsr PutString
        LoadW r0,EightRow ;Ausgabezeile 8
        LoadB r1H,168 ;y-Position
        LoadW r11,20 ;x-Position
        jsr PutString
        LoadW r0,NinthRow ;Ausgabezeile 9
        LoadB r1H,177 ;y-Position
        LoadW r11,20 ;x-Position
        jsr PutString
        rts
```

© 64'er

Datensatz aktualisieren und speichern

```
AcIcon1:        jsr UpdateRecordFile
                jsr CloseRecordFile
                jmp EnterDeskTop

allgemeiner Quelltext
AcBeenden:      jsr UpdateRecordFile
                jsr CloseRecordFile
                jmp EnterDeskTop
```

© 64'er

Neue System-Piktogramme

Folge 2

Auf der Suche nach den Geos-Icons

Jede Geos-Datei besitzt ein individuelles Icon, das man mit der Maus komfortabel wählt und per Klick aktiviert. Die Sprite-Muster dieser Piktogramme lassen sich aber jederzeit ändern – man muß nur wissen, wie!

Diesmal kümmern wir uns um die Icons von DeskTop und TopDesk. Beide Geos-Systemdateien arbeiten mit bis zu acht Piktogrammen, die jedem Geos-User vertraut sind. Die entsprechenden Fundorte der Icon-Daten zeigt Ihnen wie im letzten Teil unserer Serie unsere übersichtliche Tabelle.

Allerdings hält sich das DeskTop-File nicht immer an den normalen Bit-Musteraufbau eines Icons: 21 Zeilen mit jeweils 24 Bildpunkten (63 Byte). Der "Mülleimer" besteht z.B. aus 66 Byte (= 22 Pixelzeilen). Deshalb teilt Geos die Daten und speichert den Rest des Bit-Musters im nächsten Sektor auf Disk. Dort liegen auch die Daten des Drucker-Icons, das wiederum nur 51 Bytes groß ist.

Weitere Besonderheiten: das Laufwerks-Icon 2 wird nur dann verwendet, wenn keine Diskette im Laufwerksschacht liegt; das CBM-Icon existiert nur für Dateien im Commodore-Format (also NICHT-GEOS); das letzte Icon in Modul 1 ist das Multi-File-Piktogramm, das man beim Übertragen oder Löschen mehrerer Dateien auf dem Screen erhält.

Auf die Besonderheiten der Piktogramme von DeskTop 128 und TopDesk 128 werden wir in der nächsten Folge eingehen.

Denis Döhler/bl

Größer als ein normales Sprite: das Bit-Muster des "Mülleimers" braucht insgesamt 66 Bytes (Modul 1 von DeskTop)

Systemeigene Geos-Icons

DeskTop 64 V2.0							
Eintrag	**Modul**	**Sektor**	**Byte-Nr.**	**Objekt**	**Größe**	**Einleitungs-Bytes**	**Besonderheit**
A 1	Nr. 1	14	ab $CF	Mülleimer	66 Byte	$C2	17 Byte n.Sekt.
B 1	Nr. 1	15	ab $14	Drucker-Ic.	51 Byte	$B3	
C 1	Nr. 1	15	ab $48	Disk-Icon	63 Byte	$BF	
D 1	Nr. 1	15	ab $88	CBM-Icon	63 Byte	$BF	7 Byte n.Sekt.
E 1	Nr. 1	15	ab $C8	MultiFile-Ic.	63 Byte	keine	

DeskTop 128 V2.0							
Eintrag	**Modul**	**Sektor**	**Byte-Nr.**	**Objekt**	**Größe**	**Einleitungs-Bytes**	**Besonderheit**
G 1	Nr. 1	16	ab $FE	Mülleimer (40)	66 Byte	$C2	62 Byte n.Sekt.
H 1a	Nr. 1	17	ab $41	Mülleimer (80)	66 Byte	$C2	1.Teil
H 1b	Nr. 1	17	anschl.	Mülleimer (80)	60 Byte	$BC	2.Teil
I 1	Nr. 1	17	ab $C3	Drucker (40)	51 Byte	$B3	
J 1a	Nr.1	17	ab $F7	Drucker (80)	48 Byte	$B0	1.Teil
J 1b	Nr. 1	18	anschl.	Drucker (80)	54 Byte	$B6	2.Teil
K 1	Nr. 1	18	ab $61	LW-Icon	64 Byte	$C0	letztes Byte: $00
L 1	Nr. 1	18	ab $A2	LW-?-Icon	64 Byte	$C0	letztes Byte: $00
M 1	Nr. 1	18	ab $C3	CBM-Icon	64 Byte	$C0	Bytes v.n.Sekt.
N 1	Nr. 1	46	ab $$65	Multi-File-Ic.	63 Byte	keine	

TopDesk 64 V1.2							
Eintrag	**Modul**	**Sektor**	**Byte-Nr.**	**Objekt**	**Größe**	**Einleitungs-Bytes**	**Besonderheit**
A 2	Nr. 1	39	ab $14	LW-Icon	63 Byte	$BF	
B 2	Nr. 1	39	ab $54	Müll-Icon	63 Byte	$BF	
C 2	Nr. 1	39	ab $94	Drucker-Icon	63 Byte	$BF	
D 2	Nr. 1	59	ab $73	Multi-File-Ic.	63 Byte	$BF	
E 2	Nr. 1	68	ab $76	Namens-Ic.	63 Byte	$BF	
F 2	Nr. 6	3	ab $9A	CBM-Icon	63 Byte	$BF	
G 2	Nr. 10	3	ab $C5	Ordner-Icon	63 Byte	$03,$15,$BF	
H 2	Nr. 11	4	ab $9A	RAM-Icon	63 Byte	$03,$15,$BF	

TopDesk 128 V1.2							
Eintrag	**Modul**	**Sektor**	**Byte-Nr.**	**Objekt**	**Größe**	**Einleitungs-Bytes**	**Besonderheit**
I 2	Nr. 1	42	ab $72	LW-Icon	63 Byte	$BF	
J 2	Nr. 1	42	ab $B3	Müll-Icon	63 Byte	$BF	
K 2	Nr. 1	42	ab $F3	Drucker-Ic.	63 Byte	$BF	Bytes v. n. Sekt.
L 2	Nr. 1	65	ab $99	Multi-File-Ic.	63 Byte	$BF	
M 2	Nr. 1	74	ab $D1	Namens-Ic.	63 Byte	$BF	Bytes v. n. Sekt.
N 2	Nr. 6	4	ab $59	CBM-Icon	63 Byte	$BF	
O 2	Nr. 10	3	ab $C5	Ordner-Icon	63 Byte	$03,$15,$BF	
P 2	Nr. 11	4	ab $9A	RAM-Icon	63 Byte	$03,$15,$BF	

Auf einen Blick – alle 6510-Befehle

Egal ob Sie Assembler-Profi oder Einsteiger sind, unsere Aufstellung der C-64-Assemblerbefehle ist ein wertvolles Hilfsmittel. Sie finden neben allen Adressierungsarten auch die Anzahl der beanspruchten Bytes im Speicher und die Zyklen, die der Befehl zur Ausführung benötigt.

Adressierungsart	Assembler-Sprachformat	Op-Code	Anzahl der Bytes	Anzahl der Zyklen	beeinflußte Flags					
					N	Z	C	I	D	V
ADC	**Addition mit Übertrag**				+	+	+	–	–	+
unmittelbar	ADC #$zz	69	2	2						
Zero-Page	ADC $00zz	65	2	3						
Zero-Page,X	ADC $00zz,X	75	2	4						
Absolut	ADC $zzzz	6D	3	4						
Absolut,X	ADC $zzzz,X	7D	3	4[1]						
Absolut,Y	ADC $zzzz,Y	79	3	4[1]						
(Indirekt,X)	ADC ($zz,X)	61	2	6						
(Indirekt),Y	ADC ($zz),Y	71	2	5[1]						
AND	**logisches UND**				+	+	–	–	–	–
unmittelbar	AND #$zz	29	2	2						
Zero-Page	AND $00zz	25	2	3						
Zero-Page,X	AND $00zz,X	35	2	4						
Absolut	AND $zzzz	2D	3	4						
Absolut,X	AND $zzzz,X	3D	3	4[1]						
Absolut,Y	AND $zzzz,Y	39	3	4[1]						
(Indirekt,X)	AND ($zzzz,X)	21	2	6						
(Indirekt),Y	AND ($zzzz),Y	31	2	5						
ASL	**Byte-Rotation nach links**				+	+	+	–	–	–
Akkumulator	ASL A	0A	1	2						
Zero-Page	ASL $00zz	6	2	5						
Zero-Page,X	ASL $00zz,X	16	2	6						
Absolut	ASL $zzzz	0E	3	6						
Absolut,X	ASL $zzzz,X	1E	3	7						
BCC	**Verzeigung bei gelöschtem Übertrag**				–	–	–	–	–	–
Relativ	BCC $zzzz	90	2	2[2]						
BCS	**Verzweigung bei gesetztem Übertrag**				–	–	–	–	–	–
Relativ	BCS $zzzz	B0	2	2[2]						
BEQ	**Verzeigung falls Ergebnis Null**				–	–	–	–	–	–
Relativ	BEQ $zzzz	F0	2	2[2]						
BIT	**Speicherbits testen**				M7	+	–	–	–	M6
Zero-Page	BIT $00zz	24	2	3						
Absolut	BIT $zzzz	2C	3	4						
BMI	**Verzweigung bei Minusresultat**				–	–	–	–	–	–
Realtiv	BMI $zzzz	30	2	2[2]						
BNE	**Verzweigung falls Ergebnis ungleich Null**				–	–	–	–	–	–
Relativ	BNE $zzzz	D0	2	2[2]						
BPL	**Verzweigung bei Plusresultat**				–	–	–	–	–	–
Relativ	BPL $zzzz	10	2	2[2]						
BRK	**Unterbrechung**				–	–	–	+	–	–
Impliziert	BRK	0	1	7						
BVC	**Verzweigung falls kein Überlauf**				–	–	–	–	–	–
Relativ	BVC $zzzz	50	2	2[2]						
BVS	**Verzweigung bei Überlauf**				–	–	–	–	–	–
Relativ	BVS $zzzz	70	2	2[2]						
CLC	**Löschen des Übertrag-Flag**				–	–	0	–	–	–
Impliziert	CLC	18	1	2						
CLD	**Löschen des Dezimal-Modus**				–	–	–	–	0	–
Impliziert	CLD	D8	1	2						
CLI	**IRQ zulassen**				–	–	–	0	–	–
Impliziert	CLI	58	1	2						
CLV	**Löschen des Überlauf-Flags**				–	–	–	–	–	0
Impliziert	CLV	B8	1	2						
CMP	**Vergleich Speicher und Akku**				+	+	+	–	–	–
unmittelbar	CMP #$zz	C9	2	2						
Zero-Page	CMP $00zz	C5	2	3						
Zero-Page,X	CMP $00zz,X	D5	2	4						
Absolut	CMP $zzzz	CD	3	4						
Absolut,X	CMP $zzzz,X	DD	3	4[1]						
Absolut,Y	CMP $zzzz,Y	D9	3	4[1]						
(Indirekt,X)	CMP ($zzzz,X)	C1	2	6						
(Indirekt),Y	CMP ($zzzz),Y	D1	2	5[1]						
CPX	**Vergleich Speicher mit X-Register**				+	+	+	–	–	–
unmittelbar	CPX #$zz	E0	2	2						
Zero-Page	CPX $00zz	E4	2	3						
Absolut	CPX $zzzz	EC	3	4						
CPY	**Vergleich Speicher mit Y-Register**				+	+	+	–	–	–
unmittelbar	CPY #$zz	C0	2	2						
Zero-Page	CPY $00zz	C4	2	3						
Absolut	CPY $zzzz	CC	3	4						

[1] bei Seitengrenzen-Überschreitung 1 addieren, [2] bei Verzeigung auf gleicher Seite 1 addieren, bei Verzeigung auf unterschiedliche Seiten 2 addieren

Adressierungsart	Assembler-Sprachformat	Op-Code	Anzahl der Bytes	Anzahl der Zyklen	N	Z	C	I	D	V
DEC	**Speicher-Dekrementierung um 1**				+	+	–	–	–	–
Zero-Page	DEC $00zz	C6	2	5						
Zero-Page,X	DEC $00zz,X	D6	2	6						
Absolut	DEC $zzzz	CE	3	3						
Absolut,X	DEC $zzzz,X	DE	3	7						
DEX	**Dekrementierung des X-Registers**				+	+	–	–	–	–
Impliziert	DEX	CA	1	2						
DEY	**Dekrementierung des Y-Registers**				+	+	–	–	–	–
Impliziert	DEY	88	1	2						
EOR	**"Exklusive-oder"-Verküpfung**				+	+	–	–	–	–
unmittelbar	EOR #$zz	49	2	2						
Zero-Page	EOR $00zz	45	2	3						
Zero-Page,X	EOR $00zz,X	55	2	4						
Absolut	EOR $zzzz	4D	3	4						
Absolut,X	EOR $zzzz,X	5D	3	4[1]						
Absolut,Y	EOR $zzzz,Y	59	3	4[1]						
(Indirekt,X)	EOR ($zzzz,X)	41	2	6						
(Indirekt),Y	EOR ($zzzz),Y	51	2	5[1]						
INC	**Speicher-Inkrementierung um 1**				+	+	–	–	–	–
Zero-Page	INC $00zz	E6	2	5						
Zero-Page,X	INC $00zz,X	F6	2	6						
Absolut	INC $zzzz	EE	3	6						
Absolut,X	INC $zzzz,X	FE	3	7						
INX	**Inkrementierung des X-Registers um 1**				+	+	–	–	–	–
Impliziert	INX	E8	1	2						
INY	**Inkrementierung desY-Registers um 1**				+	+	–	–	–	–
Impliziert	INY	C8	1	2						
JMP	**Sprung zu Speicherzelle**				–	–	–	–	–	–
Absolut	JMP $zzzz	4C	3	3						
Indirekt	JMP ($zzzz)	6C	3	5						
JSR	**Sprung zu einem Unterprogramm**				–	–	–	–	–	–
Absolut	JSR $zzzz	20	3	6						
LDA	**Speicherübertragung zum Akkumulator**			+	+	–	–	–	–	
unmittelbar	LDA #$zz	A9	2	2						
Zero-Page	LDA $00zz	A5	2	3						
Zero-Page,X	LDA $00zz,X	B5	2	4						
Absolut	LDA $zzzz	AD	3	4						
Absolut,X	LDA $zzzz,X	BD	3	4[1]						
Absolut,Y	LDA $zzzz,Y	B9	3	4[1]						
(Indirekt,X)	LDA ($zzzz,X)	A1	2	6						
(Indirekt),Y	LDA ($zzzz),Y	B1	2	5[1]						
LDX	**Speicherübertragung zum X-Register**				+	+	–	–	–	–
Unmittelbar	LDX #$zz	A2	2	2						
Zero-Page	LDX $00zz	A6	2	3						
Zero-Page,Y	LDX $00zz,Y	B6	2	4						
Absolut	LDX $zzzz	AE	3	4						
Absolut,Y	LDX $zzzz,Y	BE	3	4[1]						
LDY	**Speicherübertragung zum Y-Register**				+	+	–	–	–	–
Unmittelbar	LDY #$zz	A0	2	2						
Zero-Page	LDY $00zz	A4	2	3						
Zero-Page,Y	LDY $00zz,Y	B4	2	4						
Absolut	LDY $zzzz	AC	3	4						
Absolut,Y	LDY $zzzz,Y	BC	3	4[1]						
LSR	**Byte-Rotation nach rechts**			0	+	+	–	–	–	
Akkumulator	LSR A	4A	1	2						
Zero-Page	LSR $00zz	46	2	5						
Zero-Page,X	LSR $00zz,X	56	2	6						
Absolut	LSR $zzzz	4E	3	6						
Absolut,X	LSR $zzzz,X	5E	3	7						
NOP	**Keine Operation**				–	–	–	–	–	–
Impliziert	NOP	EA	1	2						
ORA	**ODER-Verknüpfung von Speicher mit Akku**				+	+	–	–	–	–
unmittelbar	ORA #$zz	9	2	2						
Zero-Page	ORA $00zz	5	2	3						
Zero-Page,X	ORA $00zz,X	15	2	4						
Absolut	ORA $zzzz	0D	3	4						
Absolut,X	ORA $zzzz,X	1D	3	4[1]						
Absolut,Y	ORA $zzzz,Y	10	3	4[1]						
(Indirekt,X)	ORA ($zzzz,X)	1	2	6						
(Indirekt),Y	ORA ($zzzz),Y	11	2	5						
PHA	**Akkumulator auf den Stapel schieben**				–	–	–	–	–	–
Impliziert	PHA	48	1	3						
PHP	**Prozessor-Status auf den Stapel schieben**				–	–	–	–	–	–
Impliziert	PHP	8	1	3						
PLA	**Stapel-Element in Akkumulator**				–	–	–	–	–	–
Impliziert	PLA	68	1	4						
PLP	**Stapel-Element in Prozessor-Status**						vom Stapel			
Impliziert	PLP	28	1	4						
ROL	**Byte-Rotation 1 Bit nach links**				+	+	+	–	–	–
Akkumulator	ROL A	2A	1	2						
Zero-Page	ROL $00zz	26	2	5						
Zero-Page,X	ROL $00zz,X	36	2	6						
Absolut	ROL $zzzz	2E	3	6						
Absolut,X	ROL $zzzz,X	3E	3	7						
ROR	**Byte-Rotation 1 Bit nach links**				+	+	+	–	–	–
Akkumulator	ROR A	2A	1	2						
Zero-Page	ROR $00zz	26	2	5						

Adressierungsart	Assembler-Sprachformat	Op-Code	Anzahl der Bytes	Anzahl der Zyklen	N	Z	C	I	D	V
Zero-Page,X	ROR $00zz,X	36	2	6						
Absolut	ROR $zzzz	2E	3	6						
Absolut,X	ROR $zzzz,X	3E	3	7						
RTI	**Rückkehr von einer Prog.-Unterbrechung**						**vom Stapel**			
Impliziert	RTI	40	1	6						
RTS	**Rückkehr von einem Unterprogramm**				–	–	–	–	–	–
Impliziert	RTS	60	1	6						
SBC	**Addition mit Übertrag**				+	+	+	–	–	+
unmittelbar	SBC #$zz	E9	2	2						
Zero-Page	SBC $00zz	E5	2	3						
Zero-Page,X	SBC $00zz,X	F5	2	4						
Absolut	SBC $zzzz	ED	3	4						
Absolut,X	SBC $zzzz,X	FD	3	4[1]						
Absolut,Y	SBC $zzzz,Y	F9	3	4[1]						
(Indirekt,X)	SBC ($zz,X)	E1	2	6						
(Indirekt),Y	SBC ($zz),Y	F1	2	5[1]						
SEC	**Übertrags-Flag setzen**				–	–	1	–	–	–
Impliziert	SEC	38	1	2						
SED	**Dezimalmodus einschalten**				–	–	–	–	1	–
Impliziert	SED	F8	1	2						
SEI	**Interrupt sperren**				–	–	–	–	–	–
Impliziert	SEI	78	1	2						
STA	**Akkumulator in Speicher ablegen**				–	–	–	–	–	–
Zero-Page	STA $00zz	85	2	3						
Zero-Page,X	STA $00zz,X	95	2	4						
Absolut	STA $zzzz	8D	3	4						
Absolut,X	STA $zzzz,X	9D	3	5						
Absolut,Y	STA $zzzz,Y	99	3	5						
(Indirekt,X)	STA ($zz,X)	81	2	6						
(Indirekt),Y	STA ($zz),Y	91	2	6						
STX	**Akkumulator in Speicher ablegen**				–	–	–	–	–	–
Zero-Page	STX $00zz	86	2	3						
Zero-Page,Y	STY $00zz,Y	95	2	4						
Absolut	STX $zzzz	8E	3	4						
STY	**Akkumulator in Speicher ablegen**				–	–	–	–	–	–
Zero-Page	STY $00zz	84	2	3						
Zero-Page,X	STY $00zz,X	94	2	4						
Absolut	STY $zzzz	8C	3	4						
TAX	**Akkumulator in X-Register speichern**				+	+	–	–	–	–
Impliziert	TAX	AA	1	2						
TAY	**Akkumulator in Y-Register speichern**				+	+	–	–	–	–
Impliziert	TAY	A8	1	2						
TSX	**Stapelzeiger in X-Register übertragen**				+	+	–	–	–	–
Impliziert	TSX	BA	1	2						
TXA	**X-Register in Akkumulator übertragen**				+	+	–	–	–	–
Impliziert	TXA	8A	1	2						
TXS	**X-Register in Stapelzeiger übertragen**				+	+	–	–	–	–
Impliziert	TXS	9A	1	2						
TYA	**Y-Register in Akkumulator übertragen**				+	+	–	–	–	–
Impliziert	TYA	98	1	2						

Maschinensprache und Basic

Nicht immer werden Programme rein in Assembler geschrieben. Um Maschinensprache-Programme aufzurufen, stellt das C-64-Basic fünf Möglichkeiten zur Verfügung:

1. Basic-Anweisung *SYS*
2. Basic-Anweisung *USR*
3. Änderung eines *RAM*-Ein-Ausgabe-Vektors
4. Änderung eines *RAM*-Unterbrechungs-Vektors
5. Änderung der *CHRGET*-Routine

1. Durch die Basic-Anweisung *SYS <adresse>* erfolgt ein Sprung zu einem Maschinen-Programm (z.B. mit *RUN* aufführbare Maschinenprogramme). Dieses muß mit einem *RTS* enden, damit die Rückkehr ins Basic gesichert ist.

2. Die Basic-Funktion *USR(X)* startet ein Unterprogramm, dessen Start sich in der Adressen 785 und 786 (High-Low-Format) befindet. Der mit *USR* übergebene Wert X ,wird im Gleitpunkt-Akkumulator #1 (ab $61 im Speicher) an das Maschinenprogramm weitergeleitet. Auf dem gleichen Weg ist eine Rückgabe von Werten vom Assembler-Programm zu Basic vorgesehen.

3. Alle Ein-/Ausgabe- und Basic-Routinen die auf die Vektor-Tabelle des C 64 zugreifen, können durch den Programmierer für eigene Zwecke verschoben bzw. geändert werden. Jeder 2-Byte-Vektor besteht aus einer niederwertigen (LOW) und höherwertigen (High) Byte-Adresse.

Diese Vektoren lassen sich mit Hilfe der Kernal-Vektor-Routinen am sichersten ändern. Der neuen Sprungvektor zeigt auf eine vom Programmierer geschriebene Unterroutine, die die Standard-Lösung ersetzt oder erweitert.

Soll danach wieder die Orginal-Routine abgearbeitet werden, muß per *JMP*-Befehl an die Adresen gesprungen werden, die zuvor im Vektor stand. Andernfalls muß die neue Routine mit dem *RTS*-Befehl enden, damit das Program sicher zum C-64-Basic zurückkehren kann.

4. Der Hardware-Interrupt-Vektor (IRQ) kann durch den User verändert werden. Die IRQ-Routine ruft der C 64 160 Mal in der Sekunde auf und benutzt diese Routine u.a. zur Zeitberechnung und zur Abfrage der Tastatur. "Hängt" man ein eigenes Programm in diese Schleife, kann man quasi parallel zum Basic-Programm eine weitere Routine im Hintergrund laufen lassen. Bitte beachten Sie, daß bei der Vektorverbiegung der Interrupt gesperrt ist (Assembler-Befehl *SEI*)

5. Der Basic-Interpreter nutzt die *CHRGET*-Routine zum Lesen der Program-Tokens, was das Hinzufügen von neuen Basic-Befehlen zum Kinderspiel macht. Ein kleines Programm prüft ob ein neuer Befehl als nächstes abgearbeitet werden soll. Ist das der Fall, verzweit es in die eigene Unterroutine. Wenn nicht, gehts zum Basic-Interpreter. *lb*





Mit unserem PRINT-USING-Ersatz, der schnellen Directory-Routine in Basic und einer Menüabfrage per Pull-Downs peppen Sie Ihre Programme ruckzuck auf.

Formatierte Zahlenausgabe in Basic

Wenn Sie eine Zahlenkolonne in Form einer Tabelle auf den Bildschirm bringen wollen, fehlt beim C 64 der *PRINT-USING*-Befehl zur formatierten Ausgabe von Zahlen und Strings. In Listing 1 finden Sie eine kleine Routine zur Formatierung und Ausgabe von Zahlen. Dazu wandelt das vorgestellte Programm die Zahlen in Strings und emittelt die Position des "Kommas".

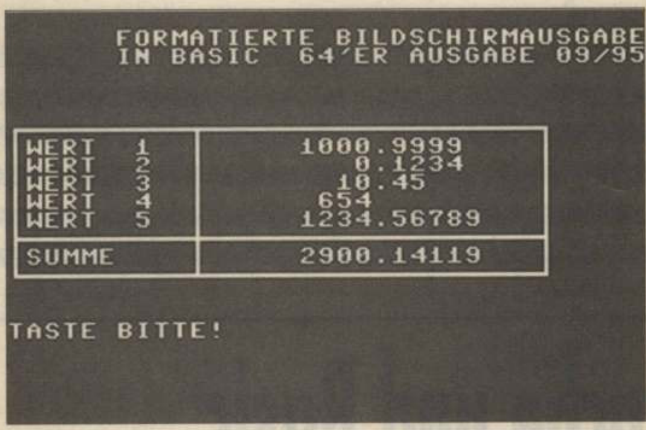
In Zeile 1010 wird die Schleife zur Bearbeitung und Ausgabe gestartet. In Zeile 1030 wird auf eine Null vor dem Komma, getestet.

Ist der Wert kleiner "1", wird durch die Anweisungen in Zeile 1040 dem Wert eine "0" vorangestellt.

In Zeile 1050 beginnt die Suche nach dem Komma im erzeugten String *A$(I)*. Dabei sucht das Programm vom "linken Ende" des Strings Zeichen für Zeichen.

Die Position des Kommas wird in der Variable *R (I)* abgelegt. Findet die Routine das Komma, verzweigt sie durch die *IF-THEN*-Anweisung in Zeile 1070 zur Bildschirmausgabe.

Die Variablen (Integer und String) in der vorgestellten Routine sind alle Feld-Elemente. Diese Methode wurde gewählt, um nach der Abarbeitung des Programms noch alle Strings und die Tabulator-Positionen verfügbar zu haben. Sie können durchaus für eine erneute Ausgabe genutzt werden. Denkbar wäre auch der Einsatz der Werte für die Ausgabe per Drucker oder zur Sicherung in einem Datenfile.



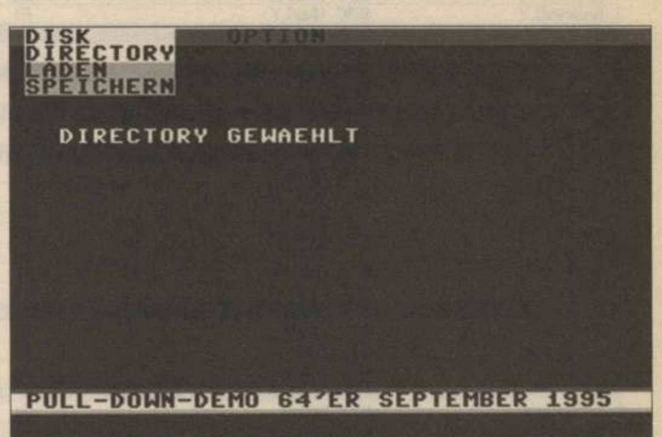
Mit Hilfe von String-Befehlen läßt sich die PRINT-USING-Funktion auch auf dem C 64 realisieren

Kurze Directory-Anzeige

Sie möchten in Ihrem Basic-Programm das Inhaltsverzeichnis einer Diskette anzeigen? Dann sollten Sie mal Listing 2 genauer unter die Lupe nehmen.In diesem kleinen Programm wird das Directory als File geöffnet, der Inhalt "String für String" eingelesen und erscheint auf dem Bildschirm. Das läuft so lange, bis die Floppy-Status-Variable *ST* gesetzt ist. Falls Sie die Directory-Anzeige als Unterprogramm aufrufen, muß in Zeile 120 für die *END*-Anweisung ein *RETURN* stehen.

Selbstverständlich können Sie auch die Nummer des Laufwerks in Zeile 20 ändern.

Der Einsatz einer Variablen ist an dieser Stelle empfehlenswert. Übrigens: Die Routine arbeitet auch mit einer Floppy 1581 bzw. FD-4000 zusammen.



Menü-Auswahl per Pull-Downs läßt sich auch in Basic programmieren - auf der Diskette zum Heft finden Sie ein Beispiel

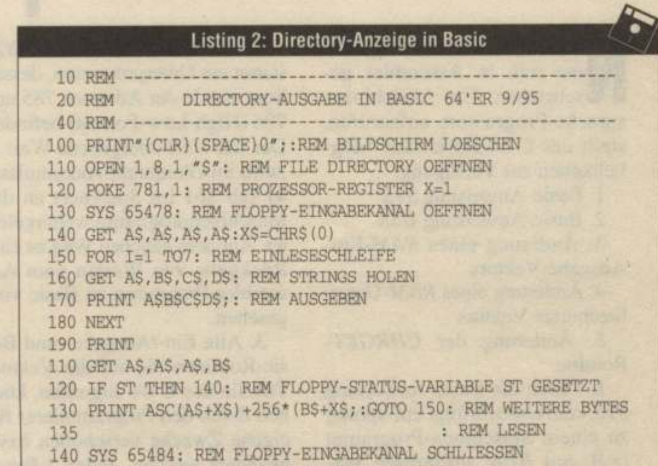
Pull-Down-Menüs in Basic

Das Programm *"PULL-DOWN-DEMO"* auf der Diskette im Heft, ist ein Grundgerüst für eine Auswahl per Pull-Down-Menüs. Sie wird per Tastatur gesteuert. Die *RETURN*-Taste bestätigt die Wahl des gewähltenMenüpunkts. Mit einem *ON-GOTO*-Befehl (s. Tips & Trick zum C 64, Ausgabe 6/1995) können Sie schnell in das gewünschte Unterprogramm verzweigen. *Jörn-Erik Burkert*

Listing 1: Formatierte Zahlenausgabe

```
1000 REM ------------------------------------------------
1000 REM BASIC-UNTERROUTINE ZUR FORMATIERTEN ZAHLENAUSGABE
1002 REM            64'ER SEPTEMBER 1995
1003 REM ------------------------------------------------
1010 FOR I=0 TO AN: REM AN=ANZAHL DER ZAHLENLEMENTE
1020 R(I)=0: REM ZAEHLER FÜR KOMMA-POSITION
1025 REM CHECK OB ZAHL GROESSER ! IST
1030 IF A(I)>1 THEN A$(I)=STR$(A(I)):L(I)=LEN(A$):GOTO1050
1035 REM ZAHLENSTRING WIRD EINE 0 VORANGESTELLT
1040 Z$=STR$(A(I)):L(I)=LEN(Z$):A$(I)="0"+(RIGHT$(Z,(L(I)-1)))
1045 REM SCHLEIFE ZUM DURCHSUCHEN DES SZTRINGS NACH DEM KOMMA
1050 FOR LL=L(I) TO 1 STEP -1
1055 REM SUCH-FUNKTION
1060 : K$=LEFT$((RIGHT$(A$(I),LL)),1)
1080 : REM IST AKTUELLES ELEMENT EIN KOMMA ?
1070 : IFK$<>"."THEN R(I)=R(I)+1:NEXT LL: REM NEIN
1075 REM KOMMA GEFUNDEN -> AUSGABE DES STRINGS
1080 PRINT TAB(TB-R(I));A$(I): REM AUSGABE TB=TABULATORWERT
1090 NEXT I: REM NAECHSTE ZAHL
1100 END
```

© 64'er



Listing 2: Directory-Anzeige in Basic

```
 10 REM ------------------------------------------------
 20 REM       DIRECTORY-AUSGABE IN BASIC 64'ER 9/95
 40 REM ------------------------------------------------
100 PRINT"{CLR}{SPACE}0";:REM BILDSCHIRM LOESCHEN
110 OPEN 1,8,1,"$": REM FILE DIRECTORY OEFFNEN
120 POKE 781,1: REM PROZESSOR-REGISTER X=1
130 SYS 65478: REM FLOPPY-EINGABEKANAL OEFFNEN
140 GET A$,A$,A$,A$:X$=CHR$(0)
150 FOR I=1 TO7: REM EINLESESCHLEIFE
160 GET A$,B$,C$,D$: REM STRINGS HOLEN
170 PRINT A$B$C$D$;: REM AUSGEBEN
180 NEXT
190 PRINT
110 GET A$,A$,A$,B$
120 IF ST THEN 140: REM FLOPPY-STATUS-VARIABLE ST GESETZT
130 PRINT ASC(A$+X$)+256*(B$+X$;:GOTO 150: REM WEITERE BYTES
135                                  : REM LESEN
140 SYS 65484: REM FLOPPY-EINGABEKANAL SCHLIESSEN
150 CLOSE 1: REM FILE SCHLIESSEN
160 END: REM BEI SUBPROGRAMM >>RETURN<<
```

© 64'er

Aus der Plus/4-Szene

Software-NEWS

Obwohl in letzter Zeit etliche User aus der Plus/4-Szene ausgestiegen sind, wird für den "harten Kern" noch immer jede Menge guter Software produziert. Wir möchten Ihnen die brandaktuellen Highlights vorstellen.

Dream World / SYNERGY: Die neue Demo von Bionic/SYNERGY stellt alles in den Schatten. Beste Effekte, z.B. Plasmas oder Dot-Tunnel, zeichnen diese Demo aus. Sie unterstützt die 256-K-Erweiterung und die SID-Card.

Signals #6/SYNERGY: Die aktuelle Ausgabe bietet wieder Infos aus der Szene, aktuelle Charts, Berichte usw. Als Bonus liegen hervorragende Demos und Anwenderprogramme bei.

10 Years Plus/4: Diese Mega-Demo wurde von vielen verschiedenen Programmierern in Ungarn hergestellt. Dort ist der Plus/4 erheblich populärer als hier in Deutschland. Das erkennt man an vielen neuen Programmen und User-Gruppen.

Cyber Shadow/TTC: Das ist eine neue Techno-Demo von Chronos mit drei Musikstücken.

Best of GFW Mega: Die Guys in West-Ungarn haben auch eine Demo "released". Einige Parts haben ausgezeichnete Effekte, wie z.B. den Amiga-Ball.

Rolling Stones Part II: Wieder ein neues Spiel aus Ungarn. Ein Boulder-Dash-Clone, der jede Menge Spaß macht.

Stone Age/Wilds: Ein Logic-Game von DCD. Es wurde von der bekannten Szene-Gruppe Wilds in Ungarn gecodet.

Thunderdome/TTC: Chronos hat eine nette One-File-Demo mit guten Grafiken herausgebracht.

Cross It II/Gentlemen Soft: Murphy/GS hat ein cooles Logic-Game gecodet, im Prinzip vergleichbar mit "Eoroid".

Gif Collection/GFW: Die GFW haben eine zweiseitige Bilder-Show herausgebracht, die sich wirklich sehen lassen kann.

Terminal Obsession/TTC: Die letzte Demo von TTC, bevor sie sich mit neuer Besetzung zu "Absence" zusammenschließen.

Logoconverter V1.0/TTC: Diese Version von Csio arbeitet mit dem "Music-Logo Maker" zusammen und konvertiert Logos in den Grafik-Screen. Eine neue Version ist schon in Planung.

Xentrix First/SYNERGY: Die erste Mega-Demo von Xentrix mit Unterstützung der SID-Card (und sogar Digi-Blaster). Leider arbeitet diese Demo nur mit der Floppystation 1541.

Explosive Net/Moldi: Ein Ungarn-Game, ähnlich wie "Minefield". Man muß man ein Feld nach Minen absuchen und sie geschickt entschärfen.

Note-Editor V1.8/TTC: Eine verbesserte Version des Noters von Chronos. Im Original wurde die C-64-Version von Cellux/Faces programmiert, jedoch hat man sie verbessert und SYNERGY-SID-Card-kompatibel gemacht.

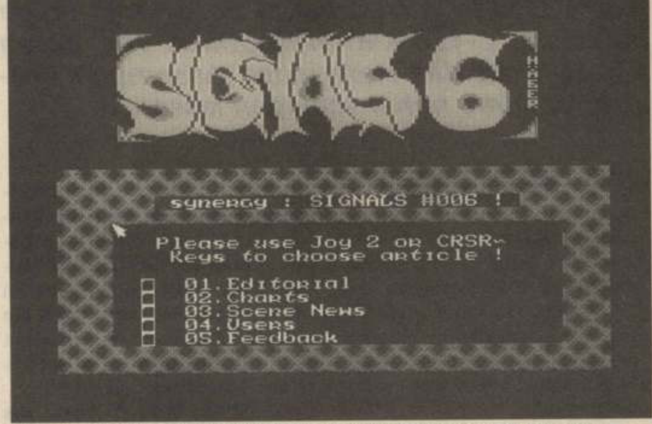
Standardseite des Disk-Magazins (hier: Szene-News): Der Herausgeber verspricht, jede Info zu veröffentlichen

Ebenfalls auf der "Signals #6"-Diskette gespeichert: der Plus/4-Bildwerfer des Grafikspezialisten Arndt Dettke

Stone Copy: Kopierprogramm mit Unterstützung der 256-K-Erweiterung für zwei Laufwerke. Kompatibel zu CMD-FD/HD 2000 & 4000, den Floppies 1551, 1541 und 1581.

Out of Time/SYNERGY & TMD: Neue Demo mit guten Effekten und SID-Card.

Magic Animation Creator V2.0/GFW: Diese zweite Version wurde entwanzt und noch verbessert. Gutes Anwenderprogramm von MAC in Ungarn.



Signals Nr. 6: Disketten-Magazin der Plus/4-Szene mit komfortablem Auswahlmenü per Joystick

Ratuchxes/Coffeine: Techno-Demo mit Torture-Effekten. Raffiniert und clever programmiert.

Andreas Friedmann/bl

Aktuelle Plus/4-Charts (Quelle: SIGNALS, Stand: 30.07.95):

Beste Gruppe:	
1. SYNERGY	70 Punkte
2. Electronic	27 Punkte
3. Scorpions	22 Punkte
4. GOTU	9 Punkte
5. Pro Pain	8 Punkte
Bester Coder:	
1. Bionic/SYNERGY	50 Punkte
2. Solder/SYNERGY	26 Punkte
3. Apos/SYNERGY	22 Punkte
4. Csory	21 Punkte
5. TCFS/Pro Pain	19 Punkte
Bester Grafiker:	
1. Hägar/SYNERGY	42 Punkte
2. Omega/Electronic	28 Punkte
3. Apos/SYNERGY	19 Punkte
4. PSP/TDC	15 Punkte
5. Unreal/Pro Pain	13 Punkte
Beste Demo:	
1. Future World/SYN	26 Punkte
2. Dream World/SYN	15 Punkte
3. Silence/SCN+U	14 Punkte
4. Infinity/EVS	10 Punkte
5. Techno Trance/GS	9 Punkte
Bestes Game:	
1. Elite/Pigmy	21 Punkte
2. Mecenary/ Novagen	20 Punkte
3. Dizzy 4/TGMS	15 Punkte
4. Digital Ball/MAD	14 Punkte
5. Eoroid Pro/DS	9 Punkte



Recht dünn gesät sind sie inzwischen – Infos, Beschreibungen oder Bücher zum C-128-Betriebssystem. Auf vielfachen Wunsch unserer Leser stellen wir ab heute in lockerer Folge nützliche C-128-Kernel-Routinen vor, deren Funktionen das Herz jedes Assembler-Programmierers höher schlagen lassen.

Der Bildschirm-Editor

Um Programm-Listings oder Daten komfortabel einzugeben, besitzt das Betriebssystem des C 128 einen "Fullscreen-Editor": man bewegt den Cursor völlig frei über den Bildschirm und ändert Programmtext oder Daten an beliebiger Stelle.

Die Funktionen des Editors beschränken sich im wesentlichen auf zwei Elemente:

- ❑ Zeichen von der Tastatur einlesen
- ❑ und auf dem Bildschirm ausgeben.

Beim C 128 kommen im Vergleich zum C 64 noch einige Features dazu, z.B. Bildschirmfenster aktivieren usw.

Wie die meisten C-128-Kernel-Routinen ruft man die Editor-Funktionen ebenfalls über Adressen einer Sprungtabelle auf (per JMP- oder JSR-Anweisung), die dann die eigentliche Routine im Betriebssystem aktivieren. Auf diese Sprungadressen des C-128-Kernel werden wir an anderer Stelle noch eingehen.

16 Funktionen umfaßt die Sprungleiste des C-128-Bildschirm-Editors:

JPCINT ($C000) Video-Chip/Editor initialisieren: setzt nach dem Einschalten bzw. nach Reset den jeweils verwendeten Video-Controller (VIC oder VDC) auf die Grundwerte zurück.

JDSPLY ($C003) Ausgabe Akku-Inhalt auf Screen, Attribut in x-Register: bringt beliebige Zeichen direkt auf den Bildschirm. Der entsprechende Code-Wert muß im Akku stehen. Den Farb-Code findet man im x-Register. Im 80-Zeichenmodus muß das Attribut-Byte zusätzlich das entsprechende Bit-Muster für Unterstreichen, Blinken usw. enthalten (= acht Bit, die sich beliebig ein- oder ausschalten lassen). Die Routine "jdsply" schreibt das Zeichen an die aktuelle Bildschirmposition (markiert durch die Speicherstellen "pnt" ($E0/E1) als Basisadresse und $EC) als Offset-Wert. Die zugehörige Adresse im Attributspeicher wird durch die Routine automatisch berechnet und steht anschließend in der Systemadresse "user" ($E2/E3).

JLP2 ($C006) Zeichen aus Tastaturpuffer in Akku: Während des Interrupts werden Tastenbewegungen lediglich registriert und im zehn Zeichen großen Tastaturpuffer zwischengespeichert.

Nach dem Aufruf der Funktion enthält der Akku das gelesene Zeichen oder eine Null (falls kein Tastendruck vorausging). Der aktuelle Inhalt des Tastaturpuffers wird um dieses eine Byte verringert, die Speicherzelle "ndx" ($D0) registriert, wieviele Zeichen gerade im Puffer sind.

Anders ist es bei den Funktionstasten F1 bis F8 sowie HELP und SHIFT+RUN/STOP: Per KEY-Anweisung kann man sie mit beliebigen Texten belegen, die nach Tipp auf die entsprechende Taste auf dem Screen erscheinen. Dazu existiert im System der Zähler "kyndx" ($D1): In dieser Adresse ist gespeichert, wieviele Zeichen auszugeben sind. Die Adresse "keyidx" ($D2) hat die Aufgabe, aufs aktuelle Zeichen innehalb der Key-Definition zu zeigen. Die Funktionstasten besitzen gegenüber dem normalen Tastaturpuffer absolute Priorität.

JLOOP5 ($C009) Zeichen vom Bildschirm in den Akku: holt ein Byte aus der gewünschten Bildschirmposition, wandelt es intern vom Screen- in den ASCII-Code und legt den Wert im Akkumulator ab. Das entsprechende Attribut-Byte steht in der Systemvariablen "tcolor" ($F2). Hier sind ebenfalls die Systemadressen $E0/E1 und $EC (s. "jdsply") im Einsatz.

JSORG ($C00F) Screen-Organisation lesen: Im Gegensatz zum C 64 lassen sich beim C 128 Bildschirmfenster (Windows) erzeugen: man kann also die maximalen Bildschirmausmaße (40 x 25 bzw. 80 x 25) beliebig verkleinern und sogar mehrere Fenster pro Screen einrichten. Mit dieser Routine verschafft man sich jederzeit Überblick, wie sich der aktuelle Bildschirm zusammensetzt: Im x-Register findet man die Anzahl der zulässigen Spalten (Zeichen pro Zeile) und im y-Register die aktuell eingestellte Zeilenzahl pro Fenster. Die Zeilenbreite (40 oder 80 Zeichen) wird im Akku abgelegt.

JKEY ($C012) Tastatur-Abfrage: identifiziert die per Tastendruck übernommenen Zeichen und stellt sie im Tastaturpuffer bereit. Die Routine ist für Programmierer unerheblich, da sie im Interrupt automatisch ausgeführt wird.

JREP ($C015) Code aus Tastaturtabelle holen, REPEAT-Test: wird ebenfalls im Interrupt automatisch generiert. Die Routine überprüft zusätzlich, ob die zuvor aktivierte Taste für den Tastatur-Wiederholmodus zugelassen und eine entsprechende Reaktion des Computers einzuleiten ist.

JPLOT ($C018) Cursor-Position lesen/schreiben: Damit läßt sich jederzeit feststellen, wo sich der Cursor auf dem Bildschirm befindet – unabhängig davon, ob er sichtbar oder unsichtbar ist. Als aktuelle Cursor-Position gilt immer die Stelle, an der per PRINT-Ausgabe das nächste Zeichen erscheinen würde.

Um die Cursor-Position zu ermitteln, muß man zunächst das Carry-Flag setzten (SEC). Nach Aufruf der Routine findet man im x-Register die aktuelle Zeilenummer und im y-Register die Spalte (beide Werte sind immer relativ zur linken oberen Ecke des Bildschirms zu sehen = Position 0,0).

Zum Schreiben der Cursor-Position ist das Carry-Flag zu löschen (CLC). Im x-Register muß ebenfalls die Zeilennummer stehen, die Spalte in y. Wurde die Assembler-Anweisung erfolgreich ausgeführt, löscht sich das Carry-Flag automatisch (bleibt es gesetzt, läßt es sich komfortabel zur Fehlerabfrage verwenden, z.B. zu hohe Zeilennummer, unzulässige Spalte usw.). Die aktuelle Zeichen-Position "pnt" ($E0/E1), der Spalten-Offset "pntr" ($EC) sowie der Zeiger auf den Attribut-Speicher "user" ($E2/E3) werden neu berechnet.

JCURS ($C01B) Cursor-Postion im 80-Zeichen-Bildschirm setzen: Der VDC-Chip erzeugt den Cursor hardwaremäßig (nicht per invertiertem Leerzeichen wie der VIC im 40-Zeichenmodus). Um die aktuelle Cursor-Position im 80-Zeichen-Screen kümmert sich die Routine "jcurs" und belegt die entsprechende Stelle des VDC-Bildschirm-RAM. Die aktuelle Zeichen-Position steht wie gewohnt in "pnt" ($E0/E1) und "pntr" ($EC).

Fürs Ein- und Ausschalten des Cursors im VDC-Screen existiert kein JMP-Aufruf in der Sprungtabelle, das muß man über die Routinen "crsron" ($CD6F) und "crsrof" ($CD9F) erledigen. Beide Routinen akzeptieren sowohl den 40- als auch den 80-Zeichenmodus (VDC-Screen).

Die Art der Cursor-Anzeige wird im 40-Zeichenbildschirm durch die Adresse "binon" ($0A26) geregelt: ist Bit 6 ($40) gesetzt, entsteht ein starrer Cursor-Block. Im 80-Zeichenmodus ist dagegen Adresse $0A2B ("curmod") zuständig. Hier kann man je nach eingeschaltetem Bit (0 bis 7) die gewünschte Cursor-Art bestimmen (s. Tabelle).

Der VDC-Cursor entsteht durch Invertieren entsprechender vertikaler Bit-Reihen eines Zeichens; die Bits 0 bis 2 bestimmen, an welcher Pixelreihe der Cursor beginnen soll: Bei "0" fängt er in der

C-128-Systemsprungtabelle Editor-Funktionen

Name	Adresse	Funktion
jpcint	$C000	Videochip/Editor initialisieren
jdsply	$C003	Ausgabe Akku auf Screen, Attribut in x-Reg.
jpl2	$C006	Zeichen aus Tastaturpuffer in Akku
jloop5	$C009	Zeichen vom Screen in Akku
jprint	$C00C	Bildschirmausgabe per Akku
jsorg	$C00F	Screen-Organisation lesen
jeky	$C012	Tastatur-Abfrage
jrep	$C015	Code aus Tastatur-Tabelle holen
jplot	$C018	Cursor-Position lesen/schreiben
jcurs	$C01B	Cursor-Position im VDC-Screen setzen
jesc	$C01E	ESC-Funktionen bearbeiten
jkyset	$C021	Funktionstaste belegen
jrsirq	$C024	Raster-Interrupt bearbeiten
jint80	$C027	80-Zeichen-Bildschirm initialisieren
jswap	$C02A	Umschatlung 40- u. 80-Zeichen-Screen
jwind	$C02D	Windoiw definieren

Cursor-Modi im VDC-Screen

Per Manipulation der Systemadresse $0A2B erzeugt man unterschiedliche Formen des Cursors im VDC-Bildschirm:

Bit Nr.	Funktion
7	nicht benutzt
6	1 = Cursor blinkt 0 = festehender Cursor
5	arbeitet eng mit Bit 6 zusammen: Bit 6 = 0, Bit 5 = 1: Cursor wird abgeschaltet Bit 6 = 1, Bit 5 = 0: schnell blinkender Cursor Bit 6 = 1, Bit 5 = 1: langsam (Normalzustand)
4	immer "0"
3	sollte stets "02 sein: der C 128 interpretiert es als höchstwertiges Bit zu den Bits 0 bis 2
0 bis 2	geben an ab welcher Pixelposition innerhalb des Zeichens der Cursor beginnt

Beispielwerte für Adresse $0A2B zur Cursor-Anzeige:

Bitbelegung	Byte-Wert	Cursor-Form
00000000	$00	fester Block-Cursor
00000111	$07	fester Unterstrich-Cursor
00100000	$20	Cursor ausgeschaltet
01000000	$40	schnell blinkender Block-Cursor
01000111	$47	Unterstrich, schnell blinkend
01100000	$60	Block-Cursor, langsam blinkend
01100111	$67	langsam blinkender Unterstrich-Cursor

ersten Reihe an (Block-Cursor), beim Bitwert 7 in der letzten (Unterstrich-Cursor). Natürlich sind beliebige Zwischenwerte möglich – nur sehen die manchmal recht seltsam aus (halber Cursor usw.).

JESC ($C01E) ESC-Funktionen ausführen: Der C 128 verfügt über eine Menge Editor-Funktionen, die per ESC-Taste eingeleitet werden. Dazu drückt man zunächst <ESC> und anschließend eine Buchstabentaste von A bis Z (s. C-128-Handbuch).

Diese Routine aktiviert ESC-Funktionen per Assembler-Programm ohne vorherigen Druck auf die ESCAPE-Taste: Man muß lediglich im Akku den ASCII-Code des gewünschten Buchstabens eintragen und die Routine per JSR aufrufen. Unsere Tabelle zeigt alle ESC-Befehle im Überblick.

JKYSET ($C021) Funktionstaste mit Text belegen: weist den Tasten <F1> bis <F8>, <HELP> und <SHIFT RUN/STOP> in Assembler-Programmen Werte zu. Die nötigen Vorbereitungen:

- In der Zeropage ist in drei aufeinanderfolgenden Adressen ein Zeiger auf den neuen Text zu richten, den man der Funktionstaste zuordnen will,
- der Akku wird mit der Adresse des ersten Bytes dieses Zeigers belegt,
- ins x-Register kommt die entsprechende Funktionstastenummer (1 bis 8, 9 = <SHIFT RUN/STOP>, 10 = <HELP>),
- das y-Register speichert die Länge des per Tastendruck auszugebenden Textes.

Dazu ein Beispiel: Sie wollen <F1> mit dem Wort "Testbelegung" definieren. Diese Zeichenkette steht z.B. ab Adresse $3000 in Bank 1. Die entsprechenden Parameter muß man in drei Zeropage-Adressen eintragen, die das Betriebssystem nie verwendet:

```
$FB = $00 (Low-Byte von $3000)
$FC = $30 (High-Byte von $3000)
$FD = $01 (Bank-Nummer)
```

Jetzt belegt man die Prozessor-Register:

```
Akku = $FB (Zeiger-Adresse)
x-Register = $01 (Nummer der
Funktionstaste)
y-Register = $0C (Textlänge)
```

Anschließend ruft man die Routine per *JSR $C021* auf.

Achten Sie beim Belegen und Definieren neuer Texte für die zehn Funktionstasten, daß der gesamte Belegungstext insgesamt nicht mehr als 248 Zeichen übersteigen darf!

JRSIRQ ($C024) Raster-Interrupt bearbeiten: Diese Routine ist für Programmierer belanglos, da sie im Interrupt automatisch aufgerufen wird. Sie löst die Umschaltung vom Text- in den Grafikbildschirm aus, wenn ein geteilter Screen per Basic-Anweisung eingestellt ist (z.B. GRAPHIC 3,1). Das Bild baut sich zeilenweise auf (eine Rasterzeile entspricht stets der Breite eines vertikalen Pixels). Der gesamte Bildschirm umfaßt theoretisch 250 Zeilen inkl. oberem und unterem Rahmen. Man kann nun den VIC-Chip veranlassen, beim Erreichen einer bestimmten Rasterzeilen-Position einen Interrupt auszulösen – solche Erreignisse werden von der Routine interpretiert und bearbeitet (Umschalten vom Grafik- in den Text-Screen, um damit einen geteilten Bildschirm zu simulieren).

JINT80 ($C027) 80-Zeichen-Bildschirm initialisieren: kopiert den ROM-Zeichensatz in Bank 14 ab Adresse $E000 in den entsprechenden Speicherbereich des VDC-RAM(ab Adresse $2000). Das passiert beispielsweise auch bei Tipp auf die <ASCII-DIN>-Taste. Das ist z.B. sehr nützlich, wenn nach eigenen Programmierexperimenten der VDC-Screen nur noch wirres Byte-Chaos zeigt oder total leer bleibt.

JSWAP ($C02A) 40/80-Zeichenbildschirm umschalten: Die Routine entspricht der ESC-Tastenfunktion "ESC X" und schaltet eine der beiden möglichen Bilddarstellungen des C 128 in (die Bildinformation kommt also wahlweise vom VIC- bzw. VDC-Controller).

JWIND ($C02D) Bildschirmfenster definieren: Damit lassen sich Windows auf dem 40- oder 80-Zeichen-Screen einstellen (s ESC-Funktionen ESC-B und ESC-T). Vor dem Aufruf muß im Akku die gewünschte Zeilennummer stehen (0 bis 24); die vorgesehene Spalte (0 bis 39 bzw. 0 bis 79) ist im x-Register zu vermerken. Bei gesetztem Carry-Flag (SEC) wird die rechte untere Windows-Ecke etabliert, bei gelöschtem Carry-Bit die linke obere.

bl

C-128-Funktionstasten (Standardbelegung)

Nummer	Taste	Belegungstext
1	<F1>	GRAPHIC
2	<F2>	DLOAD"
3	<F3>	DIRECTORY + CHR$(13)
4	<F4>	SCNCLR + CHR$(13)
5	<F5>	DSAVE"
6	<F6>	RUN + CHR$(13)
7	<F7>	LIST + CHR$(13)
8	<F8>	MONITOR + CHR$(13)
9	<SHIFT+RUN/STOP>	DLOA "*"+CHR$(13)+RUN+CHR$(13)
10	<HELP>	HELP + CHR$(13)

ESC-Funktionen per System-Routine JESC ($C01E)

Bevor man die Routine per JSR-Anweisung aktiviert, muß im Akkumulator der gewünschte Funktions-Code (ASCII-Wert) stehen:

Akku-Inhalt	Funktion
@$40)	Bildschirm ab Cursor-Position löschen
A ($41)	Auto-Insert-Modus einschalten
B ($42)	rechte untere Fensterecke an aktueller Cursor-Position setzen
C ($43)	Auto-Insert-Modus aus
D ($44)	aktuelle Zeile löschen. Der restliche Screen rollt nach oben.
E ($45)	festen Cursor setzen
F ($46)	Cursor blinkt
G ($47)	Signalton zulassen (entspricht der Basic-Anweisung "PRINT CHR$(7)".
H ($48)	Signalton abstellen. Entsprechende Aufrufe werden ignoriert.
I ($49)	Zeile an Cursor-Position einfügen. Folgende Zeilen verschieben sich nach unten.
J ($4A)	Cursor an Zeilenanfang
K ($4B)	Cursor an Zeilenende
L ($4C)	aktiviert Bildschirm-Scrolling. Erreicht man den unteren Rand per PRINT-Befehl, rollen die Zeilen nach oben und eine Leerzeile taucht unten auf.
M ($4D)	verhindert das Rollen des Bildschrims. Ereicht man bei Eingaben oder mit PRINT-Befehlen den untersten Rand, wird die Ausgabe wieder in der obersten Zeile fortgesetzt. Nachteil: Der alte Bildschirm-Inhalt wird überschrieben.
N ($4E)	Revers-Anzeige ausschalten
O ($4F)	Multifunktion: Einfüge-, Anführungszeichen- und Revers-Modus ausschalten
P ($50)	Zeile bis Cursor-Position löschen
Q ($51)	... ab Cursor-Position löschen
R ($52)	Bildschirm invertieren (gilt nicht für den Screen-Rahmen)
S ($53)	Block-Cursor einschalten
T ($54)	obere linke Windows-Ecke an aktueller Cursor-Position einrichten
U ($55)	Unterstrich-Cursor einschalten
V ($56)	Bildschirm um eine Zeile nach oben scrollen
W ($57)	... um eine Zeile abwärts
X ($58)	Umschaltung 40-/80-Zeichenmodus
Y ($59)	Standard-Tabulatoren setzen
Z ($5A)	alle Tabulatoren löschen

Von Basic zu

Im letzten Teil unseres Umsteiger-Kurses wollen wir uns mit der 16-Bit-Multiplikation bzw. Division und der Nutzung des C-64-Betriebssystems befassen.

Nachdem wir in der letzen Folge schon mit 16-Bit-Zahlen gerechnet haben, wollen wir nun unsere Ergebnisse auf den Bildschirm bringen. Dazu könnten wir eine eigene Routine entwickeln, die die Zahl in einen String im Speicher umrechnet und dann die einzelnen Bytes in den Bildschirm schreibt. Diese Arbeit können wir uns jedoch sparen, denn das Betriebssystem des C 64 birgt ein kleines Unterprogramm, das diese für uns Aufgabe übernimmt.

Bevor wir aber so eine Routine aufrufen, müssen wir einige Vorbereitungen treffen und Parameter übergeben. Ein kleines Demoprogramm im Maschinensprache-Monitor SMON soll das verdeutlichen:

1000 LDX #00
1002 LDA #20
1004 JSR BDCD
1007 RTS

Wenn Sie nun das Programm mit *SYS 4096* starten, gibt die Routine den Wert 32 auf dem Bildschirm aus.

Um eine 16-Bitzahl auf dem Screen auszugeben, wird die Betriebssystem-Routine *LINPRT* ($BDCD) aufgerufen. Zuvor muß sich der Zahlenwert im High/Low-Format im Akkumulator und im X-Register befinden (in unserem Beispiel 32).

Listing 1: Ausgabe ohne Beachtung des Vorzeichens

```
             *= $1000           ;startadresse

             sec                ;subtraktion
             lda zahl1
             sbc zahl2
             tax
             lda zahl1+1
             sbc zahl2+1
             jsr $bdcd          ;ausgabe
             rts                ;zureuck
zahl1   .byte $00,$00
zahl2   .byte $00,$08
```


Listing 2: Ausgabe mit Beachtung des Vorzeichens

```
             *= $1000           ;startadresse

             sec                ;subtraktion
             lda zahl1
             sbc zahl2
             tay
             lda zahl1+1
             sbc zahl2+1
             jsr $b391          ;ergebnis in FAC
             jsr $aabc          ;ausgabe
             rts                ;zureuck
zahl1        .byte $00,$00
zahl2        .byte $00,$08
```


Listing 3: Cursor positionieren und String ausgeben

```
             *= $1000           ;startadresse

             ldx #$05           ;spalte
             ldy #$02           ;zeile
             jsr $e50c          ;cursor setzen
             lda #<text         ;lo-byte txt-adresse
             ldy #>text         ;hi-byte txt-adresse
             jsr $able
             rts
text         .text    "64'er-magazin"
             .byte $00 ;endkennung
```


Die Routine arbeitet aber leider nur mit Zahlen im Bereich 0 bis 65535 zusammen und beachtet daher das Vorzeichen einer 16-Bitzahl überhaupt nicht.

Kursübersicht

- Teil 1: Einstieg in Assembler, erste Befehle, einfache Schleifen
- Teil 2: Adressierungsarten, Branch-Befehle, Arithmetik
- **Teil 3: Weitere Programmiertricks, Nutzung des Betriebssystems und Praxis-Anwendung**

Ein neuer Akkumulator

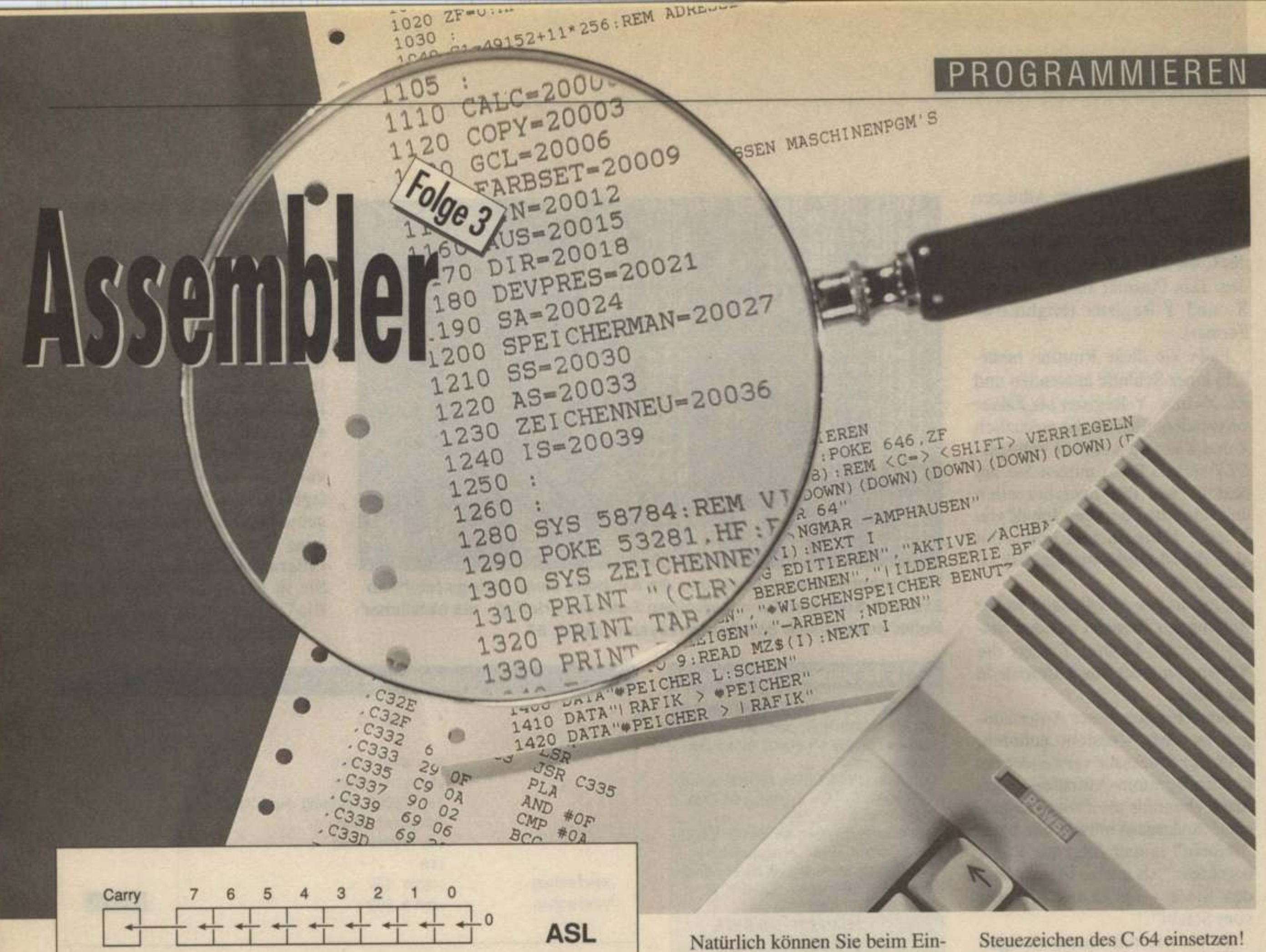
Um den Zahlenbereich auch auf die negative Skala auszudehnen, hat der C 64 zwei weitere Akkumulatoren integriert. Die Fließkomma-Akumulatoren *FAC* und *ARG* sind für "Werte mit Komma" verantwortlich und greifen auch auf den negativen Wertebereich zu, wobei Zahlen von -32768 (hex. $8000) bis 32767 (hex. $7FFF) zugelassen sind. Aus Platzgründen, wollen wir den *FAC* nur als Hilfsmittel zur vorzeichenrichtigen Ausgabe benutzen (keine Sorge ein spezieller Kurs zu diesem Thema ist schon in Vorbereitung). Listing 1 zeigt eine Subtraktion und die Ausgabe des Ergebnisses mit Hilfe der Betriebssystem-Routine *LINPRT* ($BDCD). Natürlich ist das Resultat falsch, denn $0000 minus $8000 ist nun mal -$8000!

In Listing 2 transferieren wir mit Hilfe der Routine *CIVAYF* ($B391) das Ergebnis der Subtraktion in den Fließkomma-Akkumulator *FAC* und geben ihn dann mit der Routine *FACOUT* ($BDD7) aus.

Listing 4: Multiplikationsbeispiel 1

```
             *= $1000           ;startadresse

             lda #$00           ;zaehler fuer
             sta 10er           ;werte-tabelle
             lda #$04           ;zaehler fuer
             sta stelle         ;stellen
loop1        tay
             lda zstring,y      ;zahl aus string holen
             sbc #$30           ;und vorbereiten
             tax
             ldy 10er
loop2        clc                ;zusammen zaehlen
             lda zahl
             adc wertlo,y
             sta zahl
             lda zahl+1
             adc werthi,y
             sta zahl+1
             dex
             bpl loop2
             inc 10er
             dec stelle
             lda stelle
             bpl loop1
output       ldx zahl           ;ausgabe der zahl
             lda zahl+1
             jsr $bdcd
             rts
zahlenstring .text  "12345"
zahl         .byte $00,$00
wertlo       .byte $01,$0a,$64,$e8,$10
werthi       .byte $00,$00,$00,$03,$27
stelle       .byte $00
10er         .byte $00
```



Carry 7 6 5 4 3 2 1 0 ← 0 **ASL**

0 → 7 6 5 4 3 2 1 0 → Carry **LSR**

7 6 5 4 3 2 1 0 Carry **ROL**

7 6 5 4 3 2 1 0 Carry **ROR**

Die Funktionsweise der Rotations-Befehle des C 64

Listing 5: Multiplikationsbeispiel mit Betriebssystem

```
        *= $1000 ;startadresse

        lda #<fak1      ;lo-byte faktor1
        sta $28
        lda #>fak1      ;hi-byte faktor 1
        sta $29
        lda #<fak2      ;lo-byte faktor 2
        sta $71
        lda #>fak1      ;hi-byte faktor 2
        sta $72
        jsr $b357       ;unmult aufrufen
        tya             ;y nach akku
output  ldx zahl        ;ausgabe der zahl
        lda zahl+1
        jsr $bdcd
        rts
```


Natürlich können Sie beim Einsatz von ausschließlich positiver Zahlen, die Variante aus Listing 1 benutzen, da hier der Wertebereich von 0 bis 65535 gesteckt ist.

Mit den vorgestellten Betriebssystem-Routinen dürfte es jetzt für Sie keine Affaire mehr sein, Ihre Rechenergebnisse auf den Bildschirm zu bringen.

Noch mehr Hilfe vom C 64

Die Ausgabe von Zahlenwerten haben wir im letzten Abschnitt besprochen. Für die Darstellung von Text auf dem Bildschirm kennen wir schon eine Variante - die Beschreibung des Bildschirm-Speichers. Eine wesentlich elegantere Methode ist die Arbeit mit der Betriebssystem-Routine *STROUT* ($AB1E), die einen String mit Endkennung 0 auf dem Bildschirm ausgibt. In Listing 3 finden Sie ein Anwendungs-Beispiel, das gleichzeitig eine weitere Betriebssystem-Funktion enthält. Bevor der String (ab Label *text*) auf den Bildschirm gebracht wird, positioniert das Programm den Cursor mit Hilfe von *PLOTK* ($E50C). Zur Vorbereitung werden Zeile und Spalte ins X- bzw. Y-Register geschrieben.

Übrings: die Benutzung von *STROUT* ($AB1E) ist nicht nur bequem, sondern birgt noch eine gewaltigen Vorteil: innerhalb des Strings können Sie wie in Basic die Steuezeichen des C 64 einsetzen!

Natürlich sind die vorgestellten Betriebssystem-Routinen nicht die einzigen. Wenn Sie zu diesem Thema noch mehr wissen wollen, sollten Sie im 64'er-Magazin 9/94 (S.6 "C64-Speichersafari") nachschlagen.

Das Multiplizieren in Assembler

Nachdem Sie die Addition und Subtraktion in der letzten Ausgabe schon kennengelernt haben, wollen wir uns in diesem Kapitel der Multiplikation zuwenden.

Mathematisch betrachtet, ist die Multiplikation eigentlich nur eine Abfolge mehrerer Additionen.

Listing 4 ist ein solches Beispiel. Hier wird eine Zahlenfolge aus einem String in eine 16-Bit-Zahl übertragen. Es gilt für das Beispiel folgende Formel:

`5*1+4*10+3*100+2*1000+1*10000`

Dazu finden Sie im Programm zwei Tabellen mit den High- und Low-Werten der einzelnen Stellen (1,10,100,1000 und 10000). Sie werden nacheinander zur 16-Bit-Zahl ab Label *zahl* addiert. Die vorgestellte Routine kann man aber auch auf zwei 16-Bit-Zahlen anwenden, was aber gar nicht notwendig ist!

Der C 64 bietet für die Multiplikation eine Routine im Betriebssystem. In Listing 5 finden Sie ein Anwendungsbeispiel. Die beiden

Faktoren müssen in den Adressen $28/$29 und $71/$72 stehen, dann kann die Multiplikations-Routines des C 64 *UMULT* aufgerufen werden. Das Produkt findet man im X- und Y-Register (High/Low-Format).

Falls Sie diese Routine innerhalb einer Schleife anwenden und das X- bzw. Y-Register als Zähler verwenden, müssen Sie zusätzlich *X* und *Y* vor dem Aufruf von *UMULT* retten! Dazu nutzen Sie am besten zwei frei Speicherzellen und schreiben dort den Inhalt von *X* bzw. *Y* hinein und holen ihn nach der Multiplikation zurück (s. Listing 6).

Eine andere Möglichkeit ist die Rettung der Register auf den Stapel (Stack). Listing 7 zeigt die Verwendung der Stack-Befehle in der Praxis.

Beim Einsatz dieser Anweisungen ist aber Vorsicht geboten, denn der Prozessor verwendet bei Unterprogramm-Aufrufen den Stack ebenfalls als Zwischenspeicher! Außerdem müssen Sie beim "Stapeln" immer folgende Regel beachten: "Als letztes Element auf den Stack - als erstes Element vom Stack!"

Es ist wie mit einem riesigen Bücherstapel - versucht man ein Buch aus dem unteren Bereich herauszuziehen, fällt der Stapel garantiert um!

Vom Rotieren

Eine weitere Möglichkeit der Multiplikation bietet der C 64 durch die Rotationsbefehle. Als Beispiel nehmen wir den Wert 1 in binärer Schreibweise:

%00000001

Wenn Sie nun in Gedanken das binär dargestellte Byte um eine Position nach links verschieben und dabei die Null auf der linken Seite rechts "anstückeln" ergibt das:

%00000010

Die Zahl hat sich verdoppelt! Diesen Effekt kann man auch mit dem Assembler-Befehl *ROL* erzeugen. Natürlich können Sie auch halbieren - die Anweisung heißt *ROR*. Ein Beispiel:

```
1000 LDA #06
1002 ROL
1003 ROL
1004 RTS
```

Nach dem Aufruf der Routine, steht im Akku der Wert 24. Beim Rotieren sichert der Prozessor die herausgeschobenen Bits im Carry-Flag und schiebt sie dann wieder ins Byte.

Die beiden Rotations-Befehle haben noch zwei Mitstreiter: *ASL* und *LSR* arbeiten im Prinzip wie ihre "Verwandten". Einziger Unterschied: das herausgeschobene Bit wird nicht auf der anderen Seite wieder hineingeschoben, sondern eine Null. Zum besseren Verständnis finden Sie in Abbildung 1 ein Wirkungsschema der Rotate-Befehle.

```
:101F 00 00 00 00 00 F2 62 72   .......
.D1000 1020
,1000  A2 05       LDX #05
,1002  A0 05       LDY #05
,1004  20 0C E5    JSR E50C
,1007  A9 0F       LDA #0F
,1009  A0 10       LDY #10
,100B  20 1E AB    JSR AB1E
,100E  60          RTS
-------------------------------------
,100F  44          ***
,1010  41 53       EOR  (53,X)
,1012  20 49 53    JSR 5349
,1015  54          ***
,1016  20 45 49    JSR 4945
,1019  4E 20 54    LSR 5420
,101C  45 53       EOR   53
,101E  54          ***
,101F  00          BRK
-------------------------------------
.M100F 1020
:100F 44 41 53 20 49 53 54 20   DAS IST
:1017 45 49 4E 20 54 45 53 54   EIN TES
:101F 00 00 00 00 00 F2 62 72   .......
.
```

Die Cursor-Positionierung und die Ausgabe eines Strings (steht ab $100f im Speicher) ist mit wenigen Befehlen erledigt - als nützlicher Helfer erweist sich das Betriebssystem des C 64

16-Bit-Division in der Praxis

Um die Wirkungsweise von Listing 8 zu verdeutlichen, spielen wir die ganze Sache an Hand eines Beispiels durch:
Wir teilen 20867 (hex. $5183) durch 321 (hex. $141) - Ergebnis 65 (hex. $41) Rest 2 (hex. $2).
Als erstes schreiben wir den Dividend und Divisor in die Ausgangsadressen und dann starten wir die Routine:

```
        LDA #$83  ;LO-BYTE
        STA $57
        LDA #$51  ;HI-BYTE
        STA $58
        LDA #$41
        STA $59
        LDA #$01
        STA $5A
;AUFRUF DER DIVISION WIE IN
LISTING 8
;ERGEBNIS IN $57/58
;REST IN $5C/$5D
```

In binärer Schreibweise sieht das so aus:

20867	*$57*	*%1000 0011*
	$58	*%0101 0001*
	$59	*%0100 0001*
	$5A	*%0000 0001*

Als Ergebnis findet man binär:

65	*$57*	*%0100 0001*
	$58	*%0000 0000*
Rest 2	*$5C*	*%0000 0010*
	$5D	*%0000 0000*

Die Bits im Griff

Sicher kennen Sie als Basic-Programmierer die Anweisungen *AND* und *OR*. In Assembler gibt es diese Befehle auch.

Sie heißen *AND* und *ORA*. Zusätzlich hat der C-64-Prozessor die Anweisung *EOR* integriert. Sie ist mit dem *NOT*-Befehl in Basic vergleichbar und kommt sehr oft bei der Booleschen Algebra zum Einsatz.

AND dient zum gezielten Löschen von Bits und *ORA* zum Setzen. Mit Hilfe der sogenannten Wahrheitstabelle (s. Infokasten "Die Wahrheitstabelle") ist die Funktionsweise der Bit-Befehle leicht zu verstehen. Die Bit-Befehle sind sehr nützlich, wenn man die obere oder unterte Hälfte eines Bytes (Nibble) bearbeitet.

Listing 6: Rettung von X- und Y- Register

```
                * = $20000

                stx xzwischen
                sty yzwischen
                jsr $1000 ;sprung zu listing 4
                ldx xzwischen
                ldy yzwischen
                rts
xzwischen       .byte $00
yzwischen       .byte $00
```


Listing 7: Rettung von X- und Y- Register mit Hilfe des Stapels

```
* = $20000

txa        ;x nach akku
pha        ;auf stapel schieben
tya        ;y nach akku
pha        ;auf stapel schieben
jsr $1000  ;sprung zu listing 4
pla        ;von stapel holen
tay        ;akku nach y
pla        ;von stapel holen
tax        ;akku nach x
rts
```


Die Wahrheitstabelle

AND	0	1
0	0	0
1	0	1
ORA	0	1
0	0	1
1	1	1
EOR	0	1
0	0	1
1	1	0

Dividieren in Assembler

Nun sind wir endlich gerüstet und können uns der letzten Grundrechenart zuwenden. Leider haben die Entwickler des C 64 vergessen, eine Divisions-Routine im Betriebssystem zu integrieren. Im Laufe der Zeit wurden unzählige Lösungen für die Division entwickelt.

Wir stellen Ihnen eine Routine vor, die zwei 16-Bit-Zahlen (Integer) verarbeitet. Sie gibt das Ergebnis und den Rest zurück! Listing 8 zeigt Ihnen das kleine Programm. Ein Beispiel finden Sie in unserem Infokasten "16-Bit-Division in der Praxis".

Natürlich ist die Divisions-Routine recht schwer durchschaubar. Sie können Sie aber auf jeden Fall zur Berechnung von Integers (ganze Zahlen) benutzen.

Weitere Rechnungsarten (z.B. Potenzieren oder Ziehen von Wurzeln) machen den Einsatz der Fließkomma-Arithmetik des C 64 notwendig. Wie schon einige Kapitel zuvor erwähnt, wollen wir uns mit diesem Problem in einem gesonderten Kurs in einer der nächsten 64'er-Ausgaben beschäftigen.

Floppy im Griff

Das Laden bzw. Speichern von Daten oder Programm-Teilen soll unseren Kurs abschließen. Hierbei wollen wir uns auf *PRG*-Dateien beschränken.

Um in Assembler den *LOAD*- bzw. *SAVE*-Befehl zu realisieren, gehen wir in drei Schritten vor:

1. Gerät wählen
2. Namen setzen
3. Floppy-Operation ausführen

Dabei werden wir wieder vom C-64-Betriebssystem kräftig unterstützt. In Listing 9 finden Sie ein Beispiel für *LOAD* und in Listing 10 eine *SAVE*-Variante.

An Hand der Basic-Befehle, wollen wir die Assembler-Versionen erklären:

Die Syntax *LOAD"name",8,1* teilen wir dazu in drei Teile auf:

1. ,8,1
2."name"
3.LOAD

Die *"8"* im ersten Teil ist die Nummer des Diskettenlaufwerkes und *"1"* deren Sekundäradresse. Um beide Werte festzusetzen benötigen wir drei Befehle:

```
LDX #$08
LDY #$01
JSR $FFBA
```

Die Betriebssystem-Routine *FILPAR* ($FFBA) setzt beide Parameter. Die *"1"* im Y-Register teilt dem Computer mit, daß die Datei an die Stelle des C 64 geladen wird, wo sie beim Speichern mit *SAVE* stand.

Steht an dieser Stelle eine *"0"*, wird die Datei an eine bestimmte (vom Programmierer festgelegte) Stelle des Speichers geladen. Die Adresse wird dem Computer später mitgeteilt.

Als nächstes wird der Dateiname festglegt, der als ASCII-Text im Speicher stehen muß. Dazu wird die Adresse des ASCII-Texts in X- und Y-Register geschrieben und die Länge des Namens in den Akkumulator:

```
LDX #<NAME
LDY #>NAME
LDA #(NAMEND-NAME)
JSR $FFBD          ;SETNAME

NAME       .NAME"file"
NAMEEND    .BYTE 0
```

Nun wird die *LOAD*-Anweisung in Richtung Floppy geschickt:

```
LDA #$00
LDX #<startadresse
LDY #>startadresse
JSR $ FFD5
```

Der Wert im Akku bestimmt, ob ein *LOAD*-Befehl (0) oder die Anweisung *VERIFY* (1) ausgeführt wird.

Das X- und Y-Register muß beschrieben sein, wenn an eine bestimmt Adresse geladen werden soll. Voraussetzung: bei *FILPAR* wurde die Sekundäradresse auf "0" gesetzt (s.o). Dann springen Sie nur noch zum *LOAD*-Befehl ($FFD5) und die Floppy beginnt zu "rattern"...

Analog zum Ladevorgang arbeitet die Speicherung von Daten. Studieren Sie Listing 10 und Sie werden schnell die Funktionsweise des *SAVE*-Befehls verstehen.

Wenn Sie Floppy-Routinen in eigenen Programmen verwenden, kann die Bildschirm-Meldung "SEARCHING..." o.ä. recht störend sein. Um diese Meldung auf dem Bildschirm zu verhindern, muß nur die Zeropage-Adresse 157 (hex. $9D) mit "0" beschrieben werden:

```
LDA #$00
STA $9D
```

Die beiden Befehle können gleich zu Programmstart oder vor den Floppy-Routinen im Listing stehen. Um den Originalzustand wieder herzustellen, müssen Sie den Wert 128 in die Zeropage-Adresse schreiben.

An dieser Stelle wollen wir unseren kleinen Exkurs in die Assembler-Welt abschließen. Sicher sind Sie nach der Lektüre noch kein Maschinensprache-Profi. Das kann sich aber schnell ändern, wenn Sie ein wenig experimentieren und die vorgestellten Beispiele variieren und kombinieren!

Jörn-Erik Burkert

So geht`s weiter

Sicherlich ist Ihnen schon aufgefallen, daß dieser Dreiteiler nicht alle Fragen zum Thema "Assembler und C 64" beantworten kann. Wir wollen unseren kleinen Exkurs in die Maschinensprache-Welt natürlich fortführen. Bitte schreiben Sie uns, welche Themen Sie besonders interessieren und auf welche Schwerpunkte wir eingehen sollen. Ihre Wünsche schicken Sie an:

Magna Media Verlag AG
Redaktion 64'er
Stichwort: Assembler
Hans-Pinsel-Str. 2
85540 Haar bei München

Unter allen Einsendern verlosen wir fünf 64'er-Assembler-Sonderhefte (inkl. Diskette) mit Kursen, Beispielen und Tools. Wie immer ist der Rechtsweg ausgeschlossen!

Die Flags des 6510

Flag	Bedeutung
C Carry-Flag-	wird gesetzt, wenn bei Additionen die Byte-Grenze von 255 überschritten wird
Z Zero-Flag-	ist gesetzt, wenn Ergebnis einer Operation gleich Null ist
I Interrupt-Flag-	ist gesetzt (Befehl SEI), wenn kein Interrupt zugelassen ist
D Dezimal-Flag-	ist gestzt, wennder Dezimal-Modus des 6510 aktiv ist
B Break-Flag-	wird gestzt, wenn der BRK-Befehl ausgeführt wurde
V Overflow-Flag-	zeigt an, wenn bei einer vorzeichenbehafteten 8-Bit-Addition der zulässige Bereich (-128 bis 127) überschritten wurde
N Negativ-Flag-	ist gesetzt, wenn das Ergebnis einer Operation negativ ist

Listing 8: Divisions-Routine in Assembler

```
         *= $1000 ;startadresse

         ldx #$00 ;ziel-bytes
         stx $5c  ;auf null
         stx $5d  ;setzen
         ldy #$10
;---------------------------------------------------------
loop     asl $57
         rol $58
         rol $5c
         rol $5d
         sec
         lda $5c
         sbc $59
         tax
         lda $5d
         sbc $5a
         bcc weiter
         stx $5c
         sta $5d
         inc $57
;---------------------------------------------------------
weiter   dey
         bne loop
         rts
```

© 64'er

Listing 9: DER LOAD-Befehl in Assembler

```
         ldx #$08            ;laufwerk 8
         ldy #$00            ;sek.-adresse 0
         jsr $ffba           ;fileparameter setzen
         ldx #<name          ;lo-byte von name
         ldy #>name          ;hi-byte von name
         lda #namend-name    ;string-laenge
         jsr $ffbd           ;name setzen
         lda #$00            ;load-befehl
                             ;(1=verfiy)
         ldx #$00            ;nach $4000
         ldy #$40            ;laden
         jsr $ffd5           ;load ausfuehren
         rts
;---------------------------------------------------------
name     .text "filename"
namend   .byte 0
```

© 64'er

Listing 10: Die SAVE-Funktion in Assembler

```
         ldx #$08               ;laufwerk 8
         jsr $ffba          ;fileparameter setzen
         ldx #<name         ;lo-byte von name
         ldy #>name         ;hi-byte von name
         lda #namend-name   ;string-laenge
         jsr $ffbd          ;name setzen
         ldx #$00           ;ab $4000
         ldy #$40           ;speichern
         stx $fa            ;zeiger in zero-page
         sty $fb            ;sichern
         lda $fb            ;zeiger auf zeropageadresse
         ldx #<endadresse
         ldy #>endadresse
         jsr $ffd8 ;file speichern
         rts                ;zurueck
                            ;zum hauptprogramm
```

© 64'er

Grafik-Kurs: Folge 1



Was wäre ein Ballerspiel, ein x-beliebiges, ohne das obligatorische Scrolling? Obwohl die Programmierung für den halbwegs assemblerkundigen User überhaupt kein Problem darstellt, gibt es immer noch Berührungsängste wenn Begriffe wie Parallaxing, Softscroll oder Y-Scrolling auftauchen. In unserem Mini-Kurs beseitigen wir alle eventuellen Unklarheiten und machen Sie in drei Kompakt-Lektionen zum Allround-Scrollprofi.

Zu Beginn der Scroll-Programmierung sind wie immer umfangreiche Grundlagen notwendig, ohne die Sie nicht auskommen. Nehmen Sie sich also ein wenig Zeit, um die generelle Funktionsweise eines Scrollers (sei er nur eine oder 24 Zeilen hoch) zu verstehen. Ist Ihnen das Grundprinzip erst einmal vertraut, werden Sie keinerlei Schwierigkeiten mehr haben, auch komplexere Vorhaben wie das eingangs erwähnte Ballerspiel zu programmieren.

Am Anfang steht der VIC

Zunächst einmal ist es wichtig zu wissen, daß der C 64 im VIC (Videochip) über ein Softscroll-Register verfügt, mit dem er den kompletten Bildschirminhalt pixelweise nach links oder rechts bewegen kann. Die jeweilige Position steht in den unteren drei Bit von Register *$D016*. Damit lassen sich insgesamt acht Positionen ansteuern. Um den Bildschirm weich in horizontaler Richtung zu verschieben, müssen Sie also lediglich nacheinander die Werte *$00* bis *$07* per *LDA*- und *STA*-Befehl in *$D016* schreiben. Für die Vertikale ist übrigens Register *$D011* zuständig, dazu kommen wir aber erst in der nächsten Ausgabe.

Ein Beispiel:

```
        LDA #$00
        STA $D016
        LDA #$01
        STA $D016
```

Bequemer geht es natürlich mit einer kleinen x-indizierten Schleife:

```
START   LDX #$00
LOOP    TXA
        STA $D016
        INX
        CPX #$08
        BNE LOOP
        JMP START
```

Der komplette Bildschirm wird jetzt ständig nach rechts geschoben und wieder auf den ursprünglichen Punkt gesetzt. Wenn Sie zusätzlich eine kleine Warteschleife mit dem Y-Register einbauen, wird es noch deutlicher. Optimal ist das ganze allerdings noch nicht, denn die entstandene Ruckelei ist kaum auszuhalten. Außerdem wird ständig der komplette Screeninhalt umhergeschoben und nicht nur ein von uns definierter Bereich. Dieses Problem aber lösen wir mit dem in der vorletzten Ausgabe bereits kennengelernten Raster-IRQ-Kniff (Ausgabe 7/95, S.27, "Die Sache mit dem IRQ"). Wie wir das anstellen? Ganz einfach:

Wir schalten die Adresse *$D016* lediglich in einem bestimmten Rasterbereich auf den jeweiligen Wert um. Diese Methode zeigt Listing 1.

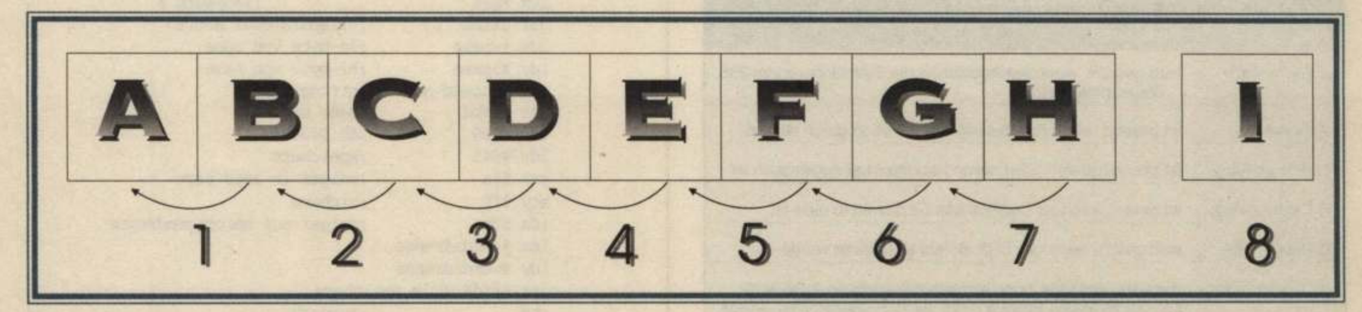
Mit Hilfe dieser kleinen Routinen wird der untere Bereich des Bildschirms ständig nach links geschoben und dann wieder in die alte Position zurückgesetzt.

Sie können das leicht überprüfen, indem Sie mit dem Cursor in den Bereich mit dem "eingeklappten" Rand fahren und einfach ein paar Zeichen auf den Bildschirm tippen. Kurz zur Erklärung:

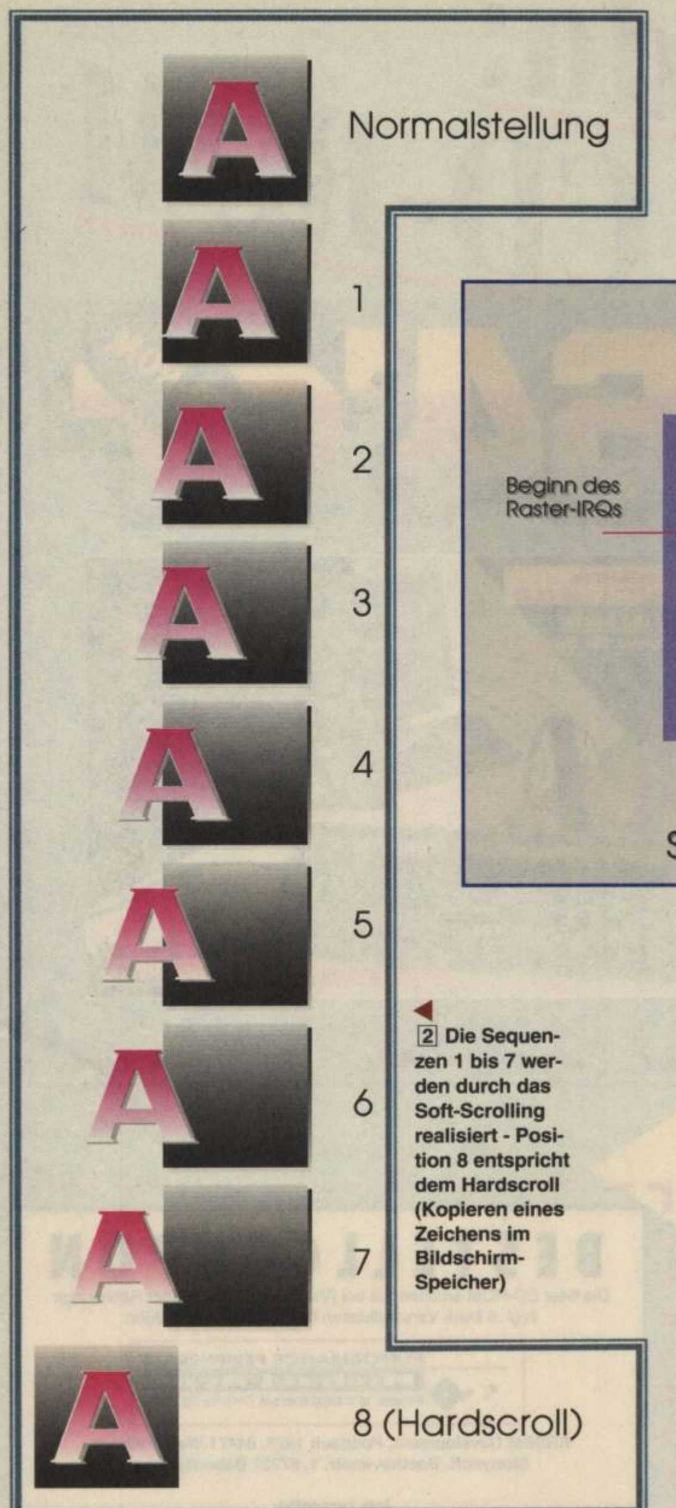
Wir setzen zunächst die zwei wichtigsten Register. Das *SCROLLHLP*-Register benötigen wir unbedingt: außerhalb des Rasterbereichs wollen wir schließlich kein Scrolling und müssen daher den Scrollregister-Wert stets auf den normalen Inhalt setzen. Deshalb nutzen wir ein Hilfsregister, wo wir stets den aktuellen *$D016*-Wert zwischenspeicheren und bei Bedarf wieder holen können. Die eigentliche Scrollroutine wird schließlich über einen einfachen *JSR SCROLL* aufgerufen. Jetzt lesen wir den aktuellen Stand des *SCROLLHLP*-Registers aus (beim ersten Aufruf enthält die Adresse den Wert *$C7*), dekrementieren den erhaltenen Wert um eins (*DEX*) und schreiben ihn in unser Hilfsregister wie auch ins eigentliche Scrollregister des VIC. Sind bereits acht Positionen geschoben, *SCROLLHLP* steht also auf *$BF*, setzen wir unser Hilfsregister wieder auf den Ursprungswert und beginnen von vorne.

Zeichen verschieben - der Hardscroll

Auch wenn Sie jetzt schon einen gehörigen Schritt weiter sind, ist das Ergebnis immer noch nicht das, was Sie sich erhofft hatten: sobald alle acht Positionen gescrollt wurden, ruckt der Bildschirmausschnitt nämlich brutal auf die ursprüngliche Stelle. Mit richtigem Scrolling, wie aus Spielen oder Demos bekannt, hat das noch nicht allzuviel gemein. Aber keine Panik: uns fehlt lediglich eine kleine Routine, die alle in der zu scrollenden Zeile stehenden Zeichen um eine Bildschirmposition nach links rückt. Das hört sich auf den ersten Blick zwar verrückt an, löst aber tatsächlich unser Problem. Abb. 1 erläutert den Vorgang genauer: sobald unsere Softscroll-Routine mit Register *$D016* alle Positionen abgeklappert hat, steht die komplette Zeile so weit links, daß ein Umkopieren der Zeichen um eine ganze Position genau einem Pixel entspricht. Das Kopieren muß dann allerdings so schnell geschehen, daß das menschliche Auge davon nichts bemerkt. Das kleine Programm in Listing 2 schiebt den Bildschirm-RAM-Inhalt von Zeile 21 beim Aufruf mit *JSR HARDSCROLL* um eine Position nach links. Natürlich können Sie auch jede andere Screenzeile damit rotieren.



[1] Die Zeichen werden auf dem Bildschirm punktweise verschoben - nach der siebenten Position müssen alle Zeichen des Scrollbereichs mit Hilfe einer Hardscroll-Routine um einen Punkt versetzt werden



2 Die Sequenzen 1 bis 7 werden durch das Soft-Scrolling realisiert - Position 8 entspricht dem Hardscroll (Kopieren eines Zeichens im Bildschirm-Speicher)

Achten Sie jedoch darauf, daß die zu verschiebende Zeile immer innerhalb unseres per Raster-IRQ eingegrenzten Bereichs liegt (sonst macht das ganze wenig Sinn). Da Sie das linke Byte nicht zwischenspeichern, geht es bei jedem Aufruf unwiederbringlich verloren. Entweder retten Sie also dieses Byte und setzen es wieder am rechten Rand (in diesem Fall an Adresse *$0747*) an, oder Sie schreiben einfach bei jedem Aufruf ein neues Byte an die rechte Zeilenposition - damit hätten Sie auch Ihren ersten "richtigen" Scrolltext".

Die komplette Hard-Scroll-Routine - mit neuem Text in der rechten Spalte - finden Sie in Listing 3. Zu Beginn setzen wir wieder ein Hilfsregister, damit wir nach dem Verlassen der Routine die zu lesende Position in unserem Scrolltext nicht verlieren und damit nicht immer der gleiche Charakter gelesen bzw. geschrieben wird.

Nach unserem Hardscroll "wissen" wir mit der Anweisung *LDY TEXTREG* an welcher Stelle des Scrolltextes wir uns gerade befinden und können somit per Y-indiziertem *LDA* das korrekte Byte an die rechte Position der Bildschirmzeile schreiben. Wurden noch nicht insgesamt 255 Zeichen gelesen, beenden wir die Routine mit *RTS*. Es ist natürlich Ihnen überlassen, wie lang Sie den Scrolltext gestalten. Beispielsweise wäre es sinnvoll, das gelesene Textbyte (steht dann im Akku) auf den Wert *$00* zu überprüfen und danach den Scrolltext wieder von vorne beginnen zu lassen. Wollen Sie mehr als 255 Zeichen Text, müssen Sie darauf achten, einen Zwei-Byte-Zähler einzuführen (Low-Byte bis *$FF* zählen lassen, Highbyte danach auf *$01* und Low-Byte wieder auf *$00* setzen).

Eingeklappter Rand

Beginn des Raster-IRQs

Ende des Raster-IRQs

Scrollbereich

3 Im mittleren Bildschirmteil (Scrollbereich), wird der Screen (per VIC-Register) für eine sanfte Bewegung des Ausschnitts "eingeklappt"

Der *INY*-Command am Ende unserer Routine ist übrigens ein kleiner Trick, der enorm Rasterzeit spart:

statt mit *LDY #$00* bzw. *LDA #$00* den Wert des *TEXTREG*isters auf Null zu setzen und damit zwei Zyklen zu verbraten, nutzen wir die Tatsache, daß der Zähler bereits auf *$FF* steht und folglich per *INY* auf *$00* gebracht werden kann. Diese Aktion benötigt nur einen Zyklus. Listing 4 zeigt noch einmal die komplette Scroll-Routine mit Rasterzeilenabfrage, Soft- und Hardscroll.

Mit diesem Assemblerprogramm können Sie jetzt endlich beliebige Bildschirmausschnitte weich verschieben. Wie groß Sie den zu scrollenden Rasterbereich machen und wieviel Bildschirmzeilen Sie insgesamt verschieben, hängt von Ihnen und der verfügbaren Rasterzeit ab.

In der nächsten Folge erfahren Sie, wie Sie in kürzester Zeit ganze Bildschirme horizontal bewegen und vertikales Scrolling realisieren. Außerdem stellen wir Ihnen die kürzeste und gleichzeitig variabelste Hardscroll-Routine vor, die uns bekannt ist.

Für angehende Profis ist der AGSP-Effekt das richtige: damit lassen sich auch HiRes-Screens in extrem wenig Rasterzeit schnell und fließend in alle Richtungen verschieben.

Peter Klein/lb

Kurs-Überblick

1. Grundlagen Scrolling
- Allgemeine Grundlagen
- Scrollen des Bildschirms
- Scrollen definierter Bereiche
- Hardscrolling

2. Scroll-Praxis
- Shoot'em-Up-Scrolling
- Vertikales Scrolling
- AGSP

3. Profi-Scrolling
- 8-Wege-Scroller
- Parallaxing
- Tips & Tricks

Listing 1: Scrollen eines Bildschirm-Abschnitts

```
SCROLLHLP       = $03FB           ;HILFSREGISTER
SCROLLREG       = $D016           ;SCROLLREGISTER

                SEI               ;IRQ SPERREN
                LDA #<START       ;LOW-BYTE NEUE IRQ-
                STA $0314 ;ROUTINE UND
                LDA #>START       ;HIGHBYTE
                STA $0315 ;SCHREIBEN
                LDA #$C7          ;SCROLLREGISTER UND
                STA SCROLLREG     ;SCROLLHELP AUF
                STA SCROLLHLP     ;NORMALWERT SETZEN
                CLI               ;IRQ FREIGEBEN
ENDLOS          JMP ENDLOS        ;ENDLOSSCHLEIFE
;-------------- RASTERZEILEN-ROUTINE --------------------
START           LDA #$B8          ;AUF RASTERSTRAHL-
LOOP1           CMP $D012 ;POSITION $B8
                BNE LOOP1 ;WARTEN
                JSR SCROLL        ;SCROLLROUTINE ANSPRINGEN
;--------------------------------------------------------
                LDA #$F0          ;AUF RASTERSTRAHL-
LOOP2           CMP $D012 ;POSITION $B8
                BNE LOOP2 ;WARTEN
                LDA #$C7          ;SCROLLREGISTER
                STA SCROLLREG     ;ZURÜCK AUF NORMALWERT
                JMP $EA31 ;ALTE IRQ-ROUTINE
;-------------- SCROLLER --------------------------------
SROLL           LDX SCROLLHLP     ;SCROLLHLPREGISTER LADEN
                DEX               ;DEKREMENTIEREN (=NACH
                STX SCROLLHLP     ;LINKS SCHIEBEN) UND
                STX SCROLLREG     ;SPEICHERN
                CPX #$BF          ;SCHON 8 POSITIONEN?
                BNE ENDSCROLL     ;NEIN DANN ENDE
                LDX #$C7          ;JA, HELP-REGISTER WIEDER
                STX SCROLLHLP     ;AUF NORMALWERT UND
                STX SCROLLREG
ENDSCROLL       RTS               ;ENDE
```


Listing 2: Die Hardscrollroutine zum Kopieren des Bildschirms

```
HARDSCROLL      LDX #$00          ;BILDSCHIRMZEILE
HARD1           LDA $0721,X       ;UM EINE
                STA $0720,X       ;POSITION
                INX               ;NACH LINKS
                CPX #$27          ;RÜCKEN
                BNE HARD1 ;NOCH NICHT DANN WEITER
;--------------------------------------------------------
                RTS
```


Listing 3: Eine komplette Scroll-Routine

```
      TEXTREG   = $02             ;HILFSREGISTER

                LDA #$00          ;HILFSREGISTER
                STA TEXTREG       ;INITIALISEREN
HARDSCROLL      LDX #$00          ;BILDSCHIRMZEILE
HARD1           LDA $0721,X       ;UM EINE
                STA $0720,X       ;POSITION
                INX               ;NACH LINKS
                CPX #$27          ;RÜCKEN
                BNE HARD1 ;NOCH NICHT DANN WEITER
;--------------------------------------------------------
                LDY TEXTREG       ;WELCHES TEXTBYTE?
                INC TEXTREG       ;TEXTBYTE-ZÄHLER ERHÖHEN
                LDA SCROLLTEXT,Y  ;BYTE HOLEN UND AN
                STA $0747 ;RECHTE POS. SCHREIBEN
                CPY #$FF          ;SCHON 255 BYTES?
                BNE HSCREND       ;NEIN DANN ENDE
;--------------------------------------------------------
                INY               ;JA, ZÄHLER AUF $00
                STY TEXTREG       ;UND SPEICHERN
HSCREND         RTS               ;ZURÜCK
;--------------------------------------------------------
SCROLLTEXT      .TEXT "SCRTEXT1"  ;MUSS 255 ZEICHEN LANG
                                  ;SEIN
```


Listing 4: Dieses Programm scrollt den Textschirm in horizontale Richtung

```
                *=$4000           ;STARTADRESSE

SCROLLHLP       = $03FB           ;HILFSREGISTER
SCROLLREG       = $D016           ;SCROLLREGISTER
TEXTREG         = $02             ;HILFSREGISTER

;--------------------------------------------------------
;------------- INITIALISIERUNG --------------------------
;--------------------------------------------------------
                SEI               ;IRQ SPERREN
                LDA #<START       ;LOW-BYTE NEUE IRQ-
                STA $0314 ;ROUTINE UND
                LDA #>START       ;HIGHBYTE
                STA $0315 ;SCHREIBEN
                LDA #$C7          ;SCROLLREGISTER UND
                STA SCROLLREG     ;SCROLLHELP AUF
                STA SCROLLHLP     ;NORMALWERT SETZEN
                LDA #$00          ;HILFSREGISTER TEXTBYTE-
                                  ;ZÄHLER
                STA TEXTREG       ;INITIALISEREN
                CLI               ;IRQ FREIGEBEN
;--------------------------------------------------------
ENDLOS          JMP ENDLOS        ;ENDLOSSCHLEIFE
;--------------------------------------------------------
;------------- HAUPTSCHLEIFE ----------------------------
;--------------------------------------------------------
START           LDA #$B8          ;AUF RASTERSTRAHL-
LOOP1           CMP $D012 ;POSITION $B8
                BNE LOOP1 ;WARTEN
;--------------------------------------------------------
                JSR SCROLL        ;SCROLLROUTINE ANSPRINGEN
;--------------------------------------------------------
                LDA #$F0          ;AUF RASTERSTRAHL-
LOOP2           CMP $D012 ;POSITION $B8
                BNE LOOP2 ;WARTEN
;--------------------------------------------------------
                LDA #$C7          ;SCROLLREGISTER
                STA SCROLLREG     ;ZURÜCK AUF NORMALWERT
                JMP $EA31 ;ALTE IRQ-ROUTINE
;--------------------------------------------------------
;------------- SCROLLER ---------------------------------
;--------------------------------------------------------
SROLL           LDX SCROLLHLP     ;SCROLLHLPREGISTER LADEN
                DEX               ;DEKREMENTIEREN (=NACH
                STX SCROLLHLP     ;LINKS SCHIEBEN) UND
                STX SCROLLREG     ;SPEICHERN
                CPX #$BF          ;SCHON 8 POSITIONEN?
                BEQ HARDSCROLL    ;NEIN DANN ENDE
;--------------------------------------------------------
ENDSCROLL       RTS               ;ENDE
;--------------------------------------------------------
HARDSCROLL      LDX #$C7          ;JA, HELP-REGISTER WIEDER
                STX SCROLLHLP     ;AUF NORMALWERT UND
                STX SCROLLREG
                LDX #$00          ;BILDSCHIRMZEILE
;--------------------------------------------------------
HARD1           LDA $0721,X       ;UM EINE
                STA $0720,X       ;POSITION
                INX               ;NACH LINKS
                CPX #$27          ;RÜCKEN
                BNE HARD1 ;NOCH NICHT DANN WEITER
;--------------------------------------------------------
                LDY TEXTREG       ;WELCHES TEXTBYTE?
                INC TEXTREG       ;TEXTBYTE-ZÄHLER ERHÖHEN
                LDA SCROLLTEXT,Y  ;BYTE HOLEN UND AN
                STA $0747 ;RECHTE POSITION SCHREIBEN
                CPY #$FF          ;SCHON 255 BYTES?
                BNE HSCREND       ;NEIN DANN ENDE
;--------------------------------------------------------
                INY               ;JA, ZÄHLER AUF $00
                STY TEXTREG       ;UND SPEICHERN
HSCREND         RTS               ;ZURÜCK
;--------------------------------------------------------
SCROLLTEXT      .TEXT "TEXT1      ;MUSS IN UNSEREM FALL 255
                                  ;ZEICHEN LANG SEIN
```



Wow! Soviel
Top Themen p
64'er
Sonderhefte
ab sofort
pro Heft nur DM 10,- (statt DM 16,-) nur solange Vorrat reicht!
Einsteiger
SH 50: Starthilfe
Alles für den leichten Einstieg/ Heiße Rythmen mit dem C64/ Fantastisches Malprogramm
SH 62: Erste Schritte
Exbasic Level II: Die Super-Basic-Erweiterung RAM-Exos: Floppy-Speeder der Extraklasse
SH 74: Einsteiger
Basic 3.5.: Basic-Erweiterung mit Hires-Grafik-Befehlen / FOBS V1.0: Floppy Operationen per Benutzeroberfläche
Anwendungen
SH 68:
Faszination Sternenhimmel. Der Blick in den Kosmos/ Kreuzworträtsel zum Selbermachen
SH 78:
Grafic-Calc:Grafische Auswertung von Jahresbilanzen/MAS V1.0: Übersichtliche Schulnotenverwaltung
SH 81:
Paint Mania: Zeichenprogramm der Superlative/ Maestro: Eigenen Sound auf Knopfdruck komponieren/ Disk Tools V6.5: Enttarnt jedes Byte auf Diskette
SH 86:
Database 2.0: Universelle Datenbank mit starken Rechenfunktionen/ Stamp Collection Kit: Archiv für Ihre Briefmarken/ außerdem: Autokauf, Haushaltskasse & Girokonto
SH 91:
RB-Fahrrad: Fahrtenbuch und Kostenverwaltung für Biker/ Video-Manager 3.0:erfaßt und verwaltet bis zu 600 Filme/ Sir-DOS V1.2: PC-Simulator für den C 64
GEOS
SH 48: GEOS-Erweiterungen
Geotext - neuer, schneller Texteditor für Geowrite/Workshop zu Geopublish
SH 59: GEOS
GeoBasic: Großer Programmierkurs mit vielen Tips & Tricks
SH 80: GEOS
Lottoblock: Statistische Gewinnzahlauswertung mit Tippvorschlägen/ Finanzen: Welche Geldanlage ist die Beste ?
SH 92: GEOS
Geo-Mensch: alles über den menschlichen Körper in Super-Grafik/ 14 optimale Druckertreiber für 24-Nadler/ 52.Master Copy: vollständiges Backup einer Geos-Disk in 52 Sekunden
SH 96: GEOS
Evolution: fantastische Reise in die Zeit der Dinosaurier und Mammuts/ Disketten-Dok: Reparatur und Sicherheitskopien beschädigter System-Disketten/Mega-Scrap: Grafikaustausch zwischen Geos- und Hi-Eddi-Format
Tips, Tricks & Tools
SH 65:
Streifzug durch die Zeropage/ Drucker-Basic: 58 neue Befehle zur Printer-Steuerung/ Multicolorgrafiken konvertieren/ über 60 heiße Tips&Tricks
SH 77:
Amica-Konvert: 6 Malprogramme tauschen Grafik aus/ Disc-Basic: Floppybefehle kurz und prägnant prüfen
SH 93:
Disk-Racoon 3.21: Diskettenverwaltung mit mausgesteuerter Benutzeroberfläche/ Basic-Expansion: 42 neue Befehle für Grafik und Floppy/ Diashow-Maker: perfektes Multimedia-Feeling
Hardware
SH 67:
Wetterstation.Temperatur, Luftfeuchtigkeit und -druck-messen/ DCF Funkuhr und Echtzeituhr/ Daten konvertieren: vom C64 zum Amiga, Atari ST und PC
SH 83: Floppy
4 Kopiertools für komplette Backups und Einzel-Files/ Spur 36 bis 40: 95 Blocks zusätzlich auf Diskette viele Tips&Tricks
SH 84: Hardware
C64 kompakt: Computer und Floppy in einem Gehäuse (Umbauanleitung)/ Midi-Interface: C64 mit Keyboard/ Tiny-EPROMer: EPROM-Brenner im Selbstbau unter 30 DM
DISKETTE IM HEFT
HARDWARE
Floppy-Speeder für 10 Mark
8-Kanal-Schaltinterface
Sieben Sinne für den C 64
Der Edel-C 64 mit Floppy
10 Bauprojekte mit Programmen
Spiele
11 Superspiele für Drahtseilnerven
Daten-Agent 00X: Jagd im All
Zoff im Geisterschloß
Charakter-Editoren: Spielelösungen, Pokes und Cheats
Alle Programme auf Diskette
Top Spiele mit Diskette
ACTION JUMP'N RUN LOGIK
Alle Hefte mit randvoller Programm-Diskette

Grafik-Tools

Studio DeLuxe

Unser Programm "Studio DeLuxe" bringt Pfiff in die Darstellung Ihrer Spiele und Anwendungen. Viele Funktionen und eine komfortable Oberfläche machen die Arbeit mit diesem Tool zum Vergnügen.

Mit "Studio DeLuxe" können Sie die 256 Zeichen eines Charsets definieren und zu einzelnen Objekten zusammenfassen. Es besteht die Möglichkeit, Hires- und Multicolor-Zeichen zu entwerfen und zu mischen. Die Zeichen-Farbe wird als Attribut gesondert im Speicher abgelegt.Die Bearbeitung des Zeichensatzes und der Objekte erfolgt mit dem Joystick in Port #2. Die restlichen Funktionen sind per Tastatur aufzurufen. Mit der Funktionstaste *<F1>* wird das Pull-Down-Menü aktiviert. Die *PFEIL-NACH-LINKS*-Taste fungiert als *"ESCAPE"* und unterbricht alle Operationen.

Die Programm-Elemente

CHARSET: Umfaßt alle 256 Charakters eines Zeichensatzes. Im rechten unteren Bildschirmteil finden Sie ein 16x16-Zeichen großes Feld, das den aktuellen Charset birgt. Über diesem Feld wird die Nummer des gewählten Zeichens und seine Farbe angezeigt. Mit dem Joystick können Sie ein Charakter wählen und Ihn danach auf dem oberen Bildschirmteil plazieren.

CURSOR: Er wird mit Hilfe des Joysticks in Port #2 oder den *CRSR*-Tasten gesteuert. Der Feuerbutton oder die *RETURN*-Taste setzen ein gewähltes Zeichen.

Das aktuelle Zeichen unter dem Cursor und dessen Farbe, finden Sie außerdem in der linken unteren Ecke des Bildschirms.

OBJECTS: Um Objekte zu wählen und zu plazieren, bewegen Sie den Cursor zum "Objekt"-Feld am linken Bildschirmrand.

Wird der Cursor auf das Modulfeld bewegt, schaltet das Programm in den Objekt-Mode. Durch Klicken auf das Feld mit "+" bzw. "-" wird in der Modul-Liste geblättert. Die gleiche Funkton erfüllen die Plus- bzw. Minus-Taste.

Die Objekt-Größe variiert zwischen 2x2, 2x3, 3x3, 3x4, 4x3 und 4x4 Zeichen pro Objekt. Die Größen sind untereinander kombinierbar und lassen sich auf Diskette sichern.

Aufbau: Object-System

Die folgende Tabelle zeigt ein Objekt im 4x4-Mode:

Chars
$5000 $5001 $5002 $5003
$5004 $5005 $5006 $5007
$5008 $5009 $500a $500b
$500c $500d $500e $500f

Attributes
$5400 $5401 $5402 $5403
$5404 $5405 $5406 $5407
$5408 $5409 $540a $540b
$540c $540d $540e $540f

Die Arbeitsmodi

CHARPUT: Hier werden Zeichen auf dem Bildschirm plaziert. Dabei steht der ganze Screen zur Verfügung - mit Ausnahme des 16x16-Zeichen-Felds. Ist dieses deaktiviert oder verschoben, können auch hier Zeichen auf den Bildschrim gesetzt werden. Zur Wahl eines Zeichens stehen vier Methoden zur Verfügung:

- im Charset-Feld (funktioniert nur bei aktivierten 16x-16-Block)
- Kopieren mit *<F7>*
- Auswahl der Chars rund um das aktuelle Zeichen im 16x16-Block mit den Funktionstasten *F3* bis *F6*
- Funktion *SCREENGET*

COLORPUT: Veränderung des Attributs des aktuellen Zeichens. Die Funktion arbeitet analog zu *CHARPUT*.



Der Arbeitsscreen von "Studio DeLuxe" - rechts die Zeichen und links die Objekte

OBJECTPUT: In diesem Mode können Sie ein Objekt aufbauen, kopiern und verschieben.Zur Aufnahme eines Objekts wechseln Sie in den Objekt-Mode und bewegen den Cursor auf die vorher gefertigte Zeichensatz-Grafik. Im Menü-Punkt "EDIT" nehmen Sie nun die Daten mit ""OBJ.READ" auf.

ZOOM: Hier lassen sich Zeichen vergrößern und editieren. Die Lupen-Funktion wird per Menü oder Z-Taste aktiviert. Der Vergrößerungsfaktor beträgt dabei acht. Nach der Aktivierung, wird mit dem Cursor der zu vergrößernde Teil gewählt und mit dem Feuerbutton bestätigt (funktioniert mit Zeichen und Objekten). Im Zoom-Mode können Sie nun Punkte setzen und löschen. Die Farben werden dabei mit den Tasten *"1"* bis *"4"* gewählt. Die *SPACE*-Taste löscht den Punkt unter dem Cursor. Verläßt der Cursor den Bildschrim, wird im Arbeitsschirm von "Studio DeLuxe" gescrollt.

Im Zoom-Mode des Tools - der Ausschnitt wir mit achtfacher Vergrößerung gezeigt

Die File-Operationen

Alle Ein- und Ausgabe-Operationen (Floppy bzw. Datasette) werden über das File-Menü abgewickelt.

DIRECTORY: Inhaltsverzeichnis holen (auch durch Taste *"D"*).

LOAD: Laden von Daten - dabei stehen folgende Dateitypen zur Auswahl (File-Kennndungen in Klammern):

- Sreen+Charset (.ALL)
- Screen (.SCR)
- Charset (.CHR)
- Objets (.OBJ)
- Animation (.ANI) - *"L"* aktiviert ebenfalls diese Funktion.

SAVE: Speichern von Daten - analgog zu LOAD (auch durch Taste (*"S"*)

COMMAND: Befehl senden

DEVICE: Angabe der Floppy- bzw. Datasetten-Nummer

EXIT: Programm beenden

Das Function-Menü

Im Function-Menü können Sie den Charset bearbeiten. Bei der Arbeit mit den Optionen gelten die Bereiche *"ONE"* (ein Zeichen), *"ALL"* (kompletter Charset) und *"AREA"* (Bereich).

INVERT: Zeichenumkehrung

FLIP X: Spiegeln um X-Achse

FLIP Y: Spiegeln um Y-Achse

SLIDE: Rotieren

CLEAR: Löschen des Screens, von Zeichen oder Objekten

BUFFER READ: Lesen eines Zeichens unter dem Cursor (arbeitet nicht im *OBJEKTPUT*-Modus)

BUFFER WRITE: Schreiben des Zeichen im Buffer an die Position unter dem Cursor

BUFFER ORA: Kombination eines Zeichens im Buffer mit dem Zeichen unter dem Cursor

CHANGE BITS: Wechseln der Bit-Kombinationen (Multicolor)

COPY: Zeichen bzw. Screenteil kopieren. Nach der Aktivierung mit dem Joystick, Cursor positionieren, mit Feuer den Mode bestätigen, obere linke Ecke wählen und dann rechten unteren Punkt markieren.

Die Editor-Einstellungen

Im Edit-Menü lassen sich einige Optionen für den Editor einstellen.

CHARSETS: Ein- bzw., Ausschalten des 16x16-Charset-Fensters

OBJECTS: Ein- bzw., Ausschalten des Objekt-Fensters

OBJECT-READ: Lesen eines Objekts aus dem Bildschirm (s. *OBJECTPUT*)

SCREEN GET: Ermöglicht das Aufnehmen eines Zeichens vom Bildschirm mit dem zugehörigen Attribut und Setzen an eine leere Bildposition.
MODE: Modi-Wahl (s. Abschnitt "Die Arbeistmodi")
SPACE: Löschen eines Zeichens (Eingabe in hexadezimaler)

Die Arbeit mit Farben

Im Color-Menü werden die Farben für Zeichen (Color), Rahmen (Border), Hintergrund (Background), Multicolor 1 (MC 1) un Multicolor 2 (MC 2) eingestellt. Dabei klappt ein Pull-Down-Menü auf und läßt Sie unter den 16 Farbtönen wählen.

Der Punkt "Multi" wechselt zwischen Hires- und Multicolor-Anzeige.

Die Speicheraufteilung

SCREEN:	
Charset	\$4000 - \$43ff
Attributes	\$4400 - \$47ff
CHARSET:	\$4800 - \$4fff
OBJECTS:	
Chars	\$5000 - \$53ff
Attributes	\$5400 - \$57ff

Zusatztools

Das Special-Menü hat einige zusätzliche Werkzeuge auf Lager:
ANIMATOR: Hier lassen sich Objekte hintereinander abspielen (Punkt *PLAY* - Stop mit Hilfe der *RETURN*-Taste). Mit *DELAY* werden die Pausen zwischen den Sequenzen festgelegt. Die Plus- bzw. Minus-Taste blättert schrittweise in der Objektliste, die mit *EDIT* bearbeitet werden kann. *STEP* bestimmt die Anzahl der Animations-Sequenzen. In *PHASES* ändert man die eigentlich Objekt-Liste.
COMPRESS: Mit dieser Funktion wird der Zeichensatz nach doppelten existierenden Chars durchsucht. Während der Suche zeigt *CHAR* das aktuelle Zeichen und *SAVE* die Anzahl der mehrfach vorkommenden Chars.
BANKS: Wechsel von Daten zwischen den internen Speicherbänken. Mit *SWAP BANKS* können die Daten zwischen den Bereichen getauscht und mit den *COPY BANK X* kopiert werden.
INFO: Programm-Information

Das Setup

Das letzte Menü hilft, daß Programm nach eigenen Vorstellungen einzurichten.
CURSOR SPEED: Cursor-Bewegungsgeschwinigkeit einstellen (bei Joystick-Steuerung)
CURSOR COLOR: Cursor-Farbe festlegen
TABLE COLOR 1: Farbe des Menü-Hintergrund wählen
TABLE COLOR 2: Farbe des Menü-Vordergrund wählen
CHARSET COLOR: Einstellung der Zeichenfarbe für den ganzen Bildschirm
GRID: Gitter in Zoom ein/ aus
GRID COLOR 1: Farbe des Grid-Gitters festlegen
GRID COLOR 2: Farbe des Grid-Gitters für Zeichen unter dem Cursor selektieren
OBJECT TYPE: Wahl der Objektgröße (2x2, 3x3 usw.)
LOAD SETUP: Laden der Einstellungen von Disk
SAVE SETUP: Einstellungen auf Diskette sichern

Die mit "Studio DeLuxe" entwickelten Gafik-Elemente können Sie in einem Level-Editor weiterverarbeiten. Wenn Sie noch nicht im Besitz eines solchen Werkszeugs sind, können Sie sich auf die nächste 64'er-Ausgabe freuen. Dann präsentieren wir Ihnen den "Level-Editor DeLuxe". Er ist zu "Studio DeLuxe" kompatibel und verarbeitet die Grafiken ohne Probleme. *Jörn-Erik Burkert*



Um gute Kompressions-Ergebnisse zu liefern, arbeiten effiziente C-64-Packer oft Stunden. Sie benötigen nur eine RAM-Erweiterung und "HACKPACK 4.0" – schon können Sie Programme in kürzester Zeit zusammenquetschen.

Das Programm "HACKPACK 4.0" von *1WAY* enpuppt sich nach dem Start als "Doppeldecker", denn es stehen gleich zwei Packer-Tools bereit. Um die Werkzeuge aber zur Arbeit zu bewegen, benötigen Sie eine RAM-Expansion 1764 bzw. 1750 von Commodore. Sollten Sie keine Speichererweiterung in Ihren C 64 eingesteckt haben, stürzt das Programm spätestens beim ersten Packvorgang ab!

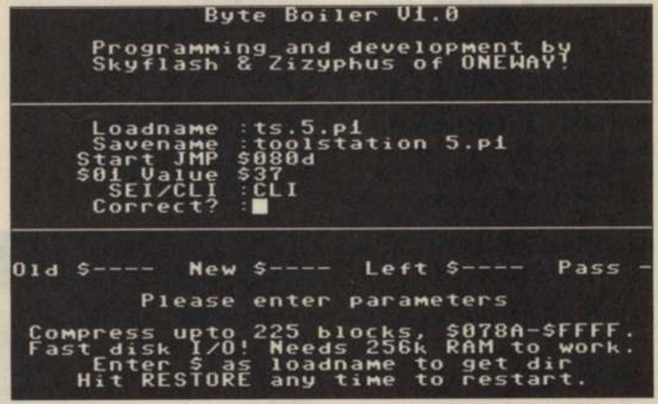
Im Intro-Screen wählen Sie per Taste einen der beiden Packer. Durch die nun folgende Kurzanleitung können Sie sich mit Hilfe der Cursor-Tasten scrollen. Per SPACE-Taste verläßt man sie und wechselt in den Arbeitsschirm.

Der ByteBoiler cruncht bis zu 200 Block lange Programme in ca. zwei Minuten. Bei den Ergebnissen steht er den bisherigen Spitzenreitern (z.B. "Timecruncher 5.0") in nichts nach - im Gegenteil es werden noch bessere Ergebnisse erzielt. Beim Einsatz eines C 128, wird der 2-MHz-Modus genutzt und damit die Packzeit fast halbiert.

ByteBoiler V1.0

Im Hauptmenü des ByteBoilers geben Sie zuerst den Namen des zu packenden Programms ein ("\$" holt das Directory von Floppy 8).

Nun folgt die Bezeichnung, unter der das gepackte File gesichert werden soll. Anschließend wird die Startadresse des gepackten Programms vermerkt. Bei "\$01 Value" wird der Original-Zustand der Speicherstelle 1 bei Programmstart angegeben. Mit Hilfe dieser Option können Sie z.B. Programme, die unter dem Betriebsystem liegen, starten. Die letzte Angabe bezieht sich auf den Zustand des Interrupt-Flaggs. Soll bei Programmstart keine System-Unterbrechung stattfinden, müssen Sie <S> für SEI wählen. Diese Option ist besonders wichtig, wenn der I/O-Bereich des C 64 (ab \$d000 bis \$dfff) oder das Betriebssystem bei Programmstart ausgeblendet ist. Zum Abschluß wird mit der Y-Taste quittiert und das Programm lädt das zu packende File. Ein Restart ist jederzeit durch die RESTORE-Taste möglich.



Der ByteBoiler packt blitzschnell und mit hervorragenden Ergebnissen - Voraussetzung ist eine RAM-Erweiterung

Nachdem die gewünschte Datei im Speicher steht, checkt der ByteBoiler die Daten und beginnt mit dem Packvorgang. Ist er beendet, kann das verkleinerte File per SPACE-Taste gesichert werden. Mit <RESTORE> springt der ByteBoiler V1.0 zum Start zurück.

AB 2.0 Level-Cruncher

Dieser Packer hilft beim Komprimieren von Files, die von einem Hauptprogramm nachgeladen werden sollen.

Das Werkzeug packt die einzelnen Dateien und schreibt sie unter demselben Namen wieder auf Diskette zurück. Mit einem Depacker werden später die Files beim Laden wieder in ihren Originalzustand versetzt und in den Speicher des C 64 befördert. Den Depacker generiert der "AB 2.0 Level-Cruncher" unter "DEP" per Befehl (*Pfeil nach links*). Dabei läßt sich die Startadresse frei bestimmen.

Der Depacker benutzt die Zeropage-Adressen *\$FB* bis *\$FF* bei seiner Arbeit. Bevor Sie ihn aufrufen, müssen Sie die Routinen SETNAM (\$ffba) und SETFLS (\$ffbd) aufrufen. Dann wird die Depack-Routine statt des LOAD-Befehls per JSR angesprungen (s. dazu den Artikel "Von Basic zu Assembler"). Bitte beachten Sie, daß die vorgestellte Software Shareware ist. Wenn Ihnen das Programm gefällt und Sie es häufig nutzen, schicken Sie 20 US-Dollar an den Autor. Sie erhalten dafür Support und Updates.

Jörn-Erik Burkert

***Info:** Dan Hovang, Halalid 26, 254 40 Helsingborg, Schweden*

Assembler-Tools

Toolstation 5.1: Neues vom ASS-Blaster

Nach diversen Updates und einer ASS-Blaster-Version für die C-64-Turbo-Karte "Flash 8", überrascht ASS-Blaster-Autor Maxim Szenessy nun mit einem gewaltigen Software-Paket rund um den "ASS-Blaster".

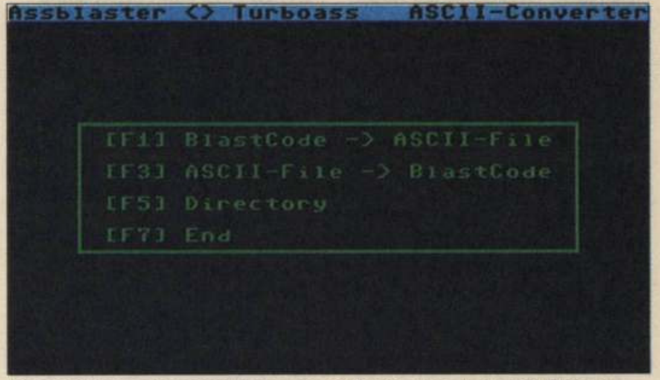
Die Zeiten, als Assembler-Programmierer mit popeligen Tools ihre Projekte abwickelten, ist schon lange vorbei. Vielmehr ist das Zeitalter des Komforts bei der Maschinensprache-Programmierung angebrochen. Die Assembler-Tools werden immer schneller, leistungsfähiger und bedienungsfreundlicher. Der "ASS-Blaster" von Maxim Szenessy ist ein Programm aus dieser Sparte. Das Programm-Paket "Toolstation 5" beinhaltet nicht nur die neueste ASS-Blaster-Version, sondern zudem viele hilfreiche Tools.

Der neue ASS-Blaster

Der Assembler ist in der Version 3.1 auf der Heft-Diskette zu finden. Er verarbeitet mittlerweile 7500 Quellcode-Zeilen und die maximale Assemblierdauer beträgt ca. 100 Sekunden (abhängig von Makro- und Kommentar-Dichte).

"ASS-Blaster 3.1" ist voll Makro-fähig. Die Bedienung erfolgt wie gewohnt wahlweise per Keyboard, Joystick oder Maus (1351 oder kompatibel).

Außerdem arbeitet der Assembler mit zwei Disketten-Laufwerken zusammen, unterstützt illegale Opcodes und verträgt sich mit den gängigen Multifunktionsmodulen (Magic Formel, Action Replay). Die Tool-Oberfläche ist konfigurierbar und die Einstellungen lassen sich auf Diskette sichern.



Konvertierung von Quelltexten vom Turbo-Ass- ins ASS-Blaster-Format und umgekehrt ist dank "Toolstation 5" ein Kinderspiel

Die "kleinen" ASS-Blaster

Zusätzlich zur neuen ASS-Blaster-Version, findet man im Software-Paket einige "abgespeckte" und an verschiedene Speicherbereiche ($4000, $5000 und $6000) angepaßte ASS-Blaster-Varianten. Ihr Funktionsumfang ist nicht so mächtig wie beim großen Bruder. Auch wurde auf Steuerung per Maus bzw. Joystick verzichtet.

Der F8-ASS-Blaster-light

Alle 16-Bit-Fans kommen bei diesem Assembler in den Genuß des vollen ASS-Blaster-Komforts. Das Tool versteht alle Befehle des Original-C-64-Prozessors und die der Turbo-Karte "Flash 8". Programmierer ohne Turbo-Karte haben ebenfalls keine Schwierigkeiten, das Programm zu nutzen und Flash-8-Software zu entwickeln. Der Assembler läuft unabhängig von der Turbo-Karte! Die Flash-8-RAM-Erweiterung wird noch nicht unterstützt.

```
AssBlasterReAss                    CODE:Mr.Lee

F1  -  Directory
F3  -  Send Disccommand
F5  -  Reassemble Objectfile
F7  -  Leave
```

Der Reassembler im Toolstation-Paket übersetzt lauffähige Maschinensprache-Programme in lesbaren Quelltext für den ASS-Blaster

Die Zusatz-Tools

Damit Umsteiger nicht in nächtelangen Sitzungen ihre alten Quelltexte erneut eintippen müssen, findet man einige Source-Code-Konverter.

Mit *"VISASS>BLASTER"* werden die Quelltexte vom VIS-ASS-Format (ASS-Blaster-Vorgänger), ins für den ASS-Blaster verständliche Format übertragen.

Gleich in beide Richtungen übersetzt *"TURBO<>BLASTER"*. Um Turbo-ASS-Quelltexte zu konvertieren, muß die Datei als sequentielles File vorliegen. Deshalb lassen sich auch SEQ-Dateien, die durch den INPUT-ASS erzeugt wurden, übersetzen. Bei umgekehrter Portierung werden SEQ-Dateien angelegt.

Der Konverter *"BLASTER>F8-BLASTER"* paßt die Quellfiles des ASS-Blaster an das erweitere Format der Flash-8-Version an. Der Quelltext-Aufbau beider Assembler ist leider nicht identisch, da der ASS-Blaster für die Turbo-Karte 24-Bit-Zahlen verarbeiten muß. Außerdem beherrscht der Prozessor der Turbo-Karte einige Adressierungsarten mehr.

Mit *"ASSBL.REASS V1.4"* läßt sich aus lauffähigen Maschinensprache-Programmen lesbarer Quellcode erzeugen, der mit dem ASS-Blaster weiterverarbeitet werden kann.

Der Toolstation-Loader

Für alle Direkt-Modus-Muffel hat der Cascade-Mitbegründer Markus Themeliden (Zeldin) ein Load-Menü integriert.
Es wird mit:
`LOAD"-*",8,1`
geladen und mit <RUN> gestartet. Im folgenden Menü können Sie die Einzel-Files per Tastendruck wählen. Wenn Sie einen Hardware-Speeder (parallel oder Modul) installiert haben, sollte der interne Software-Beschleuniger auf "OFF" stehen. Ist das nicht der Fall, stürzt der Loader beim nächsten Diskettenzugriff ab.

Wer zahlt, gewinnt

Wenn Sie das Programm häufig nutzen wollen, sollten Sie sich beim Autor als Nutzer eintragen lassen. Die Registratur-Gebühr beträgt 20 Mark. Als eingetragener User erhalten Sie Support und Updates zur Toolstation.

Jörn-Erik Burkert

Info: *Maxim Szenessy, Brunnenweg 12, 25436 Tornesch*

Die Files der Toolstation 5.1

File	Bedeutung
-TOOLSTATION5.1-	Loader
READ THIS!	Info-Note 1
ASSBLASTER V3.1	ASS-Blaster C 64/128
BED	
BE+	
BT+	
ASSBLASTERPREFS	ASS-Blaster-Einstellungen
MAKR.XAMPLES.SCR	Quelltext-Beispiel für Makros
ASSBL.V3-DOC.SCR	Anleitung zum ASS-Blaster V 3.1
ASSBL.V3.1-$4000	ASS-Blaster-Version ab $4000
ASSBL.V3.1-$5000	ASS-Blaster-Version ab $5000
ASSBL.V3.1-$6000	ASS-Blaster-Version ab $6000
F8-ASS-Blaster V1	ASS-Blaster für die C-64-Turbokarte "Flash 8"
TURBO<>BLASTER	Source-Code-Konverter für Turbo-ASS (Omikron) und ASS-Blaster
VISASS>BLASTER	Source-Code-Konverter für VIS-ASS zu ASS-Blaster
ASSBLAST>FLASH8	Source-Code-Konverter für ASS-Blaster zu Flash-8-ASS-Blaster
ASSBL.REASS V1.4	Reassembler - erzeugt Quellcode für den ASS-Blaster
ASSBL.INFONOTE	Infonote 2

Das deutschsprachige Disketten-Magazin "Digital talk" feierte unlängst seinen zweiten Geburtstag. Grund genug für uns, die Nase etwas tiefer in die Ausgabe 15 zu stecken.



Nach dem Laden und dem Start von "Digital Talk" (DT) erkennt man recht schnell den Unterschied zu den üblichen Disk-Mags – alle Texte sind in deutsch! Bevor man aber direkt ins Hauptmenü gelangt, kann man das Diskettenmagazin an die eigene Computeranlage anpassen. Der Programmierer Tobias "Dr. Zoom" Erbsland hat cleverweise alle möglichen Hardware-Erweiterungen bei der Konzeption von DT berücksichtigt: u.a. RAM-Erweiterungen, Fastloader, GEO-RAM und die Flash-8-Turbokarte.

Im Hauptmenü angelangt, werden die einzelnen Unterrubriken per Joystick gewählt. Mit dem Steuerhebel blättert man auch in den Kapiteln. Proportinalschrift und integrierte Grafiken sind Markenzeichen von DT und geben dem Magazin ein tolles Outfit. Beim Blättern fallen leider sehr oft "zerstückelte" Sätze auf, deren Fragmente auf zwei Seiten verteilt sind. Hier sollte die Redaktion ein wenig mehr auf's Layout achten und das Mag ein bißchen lesefreundlicher gestalten.

Neuigkeiten und Gerüchte

Einen breiten Raum in DT nimmt der Aktuell-Teil ein. Hier findet der Leser ein Editorial, Nachrichten und Reaktionen auf Leserbriefe. Der News-Teil befaßt sich weniger mit üblichem Szene-Klatsch, sondern bringt echte Neuigkeiten aus der Computer-Welt und schaut dabei auch ein wenig über den Tellerrand.

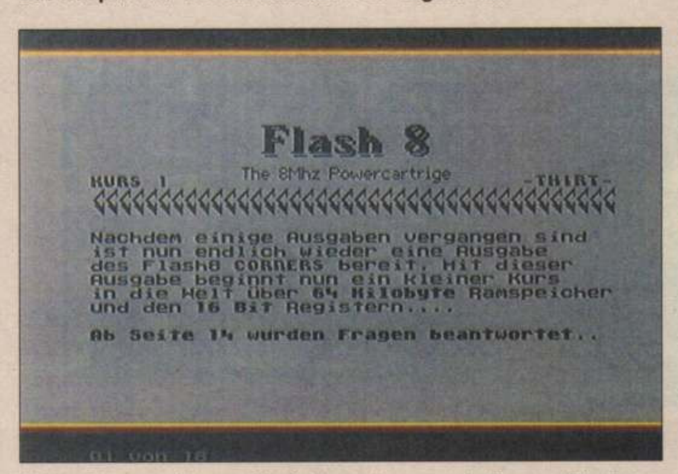
Viel Platz werden den Leser-Diskussionen und Stellungsnahmen eingeräumt. Jeder Leser kann seinen Beitrag zur DT an die Redaktion schicken.

Service und Programmieren

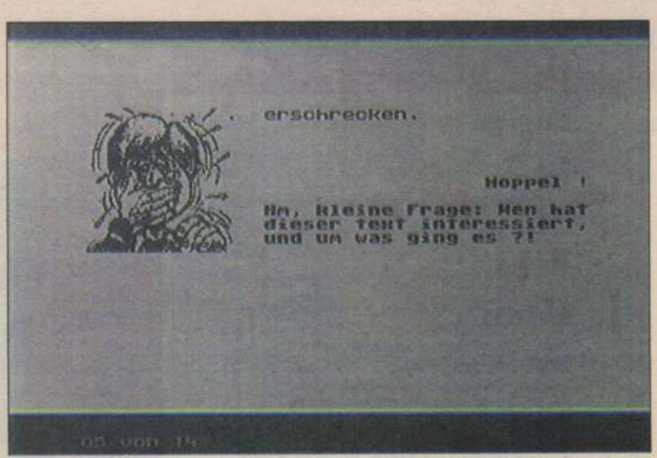
Eine weitere sehr umfangreiche Rubrik ist "6510 inside". Hier dreht sich alles um den C 64! Mit Assembler-Kursen, Beiträgen zu Geos oder Flash-8 geben die User anderen Freaks Praxistips und helfen Neueinsteigern beim Programmieren in Assembler. In der Hardware-Corner stapeln sich nützliche Bastelanleitungen. In der Abteilung "Buchtip", kramt die Redaktionscrew diverse Oldies aus dem Keller und geben Tips, wie man sich ein schon lang vergriffenes Meisterwerk an Land zieht. Spieler finden in diesem Kapitel kritische Tests aktueller Games.



Das Hauptmenü und die Rubriken von "Digital Talk"



Neben Geos und Assembler bekommt man auch Tips zur Flash-8



Das Magazin-System erlaubt auch Grafiken zwischen dem Text

Leute, Interviews und Adressen

In der Rubrik "Interviews und Steckbriefe" stellen die DT-Macher schillernde C-64-People vor. Natürlich findet man die Entwicklungsgeschichte der Freaks, Infos über die Leibspeise oder über die aktuelle Lieblingsband. Außerdem quetscht die Redaktion die Interview-Partner nach ihrer Meinung zu aktuellen Themen aus.

Der (leider) etwas zu knapp geratene Adressenteil vermittelt schnell Kontakte zu anderen C-64-Fans.

Unterhaltung und Spaß

Natürlich hat sich die DT-Crew nicht nur die C-64-Computerei auf die Fahnen geschrieben, sondern auch noch viele andere interessante Stories in petto. In "Moviecorner" findet der Leser so manchen Film-Geheimtip. Dabei wird nicht auf bekannte Kommerzschinken eingegangen, sondern die Autoren beschäftigen sich mehr mit unbekannten, aber nicht weniger interessanten Zelluloid-Werken.

Zusätzlich veröffentlicht DT Leser-Beiträge zu popuärwissenschaftlichen Themen wie UFOs oder Mythologie und Berichte zu Kultur, Reisen, Konzerten oder auch witzige Kurzgeschichten.

Jörn-Erik Burkert

Die Mag-Macher

Die Redaktion von "Digital Talk" besteht aus den beiden Chefredakteuren Jörg Nehls (Deutschland) und Johannes Rinderer (Österreich). Außerdem gehört der Herausgeber und Mag-Programmierer Tobias Erbsland (Schweiz) zum Team von "DigitalTalk".

Natürlich kann jeder Leser bei der Gestaltung mitmischen. Die DT-Crew freut sich über jede Einsendung - Briefe und Beiträge können Leser auch auf Papier an die Redaktion schicken.

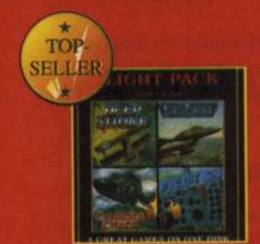
Eine andere Möglichkeit ist der DT-Editor. Mit dessen Hilfe kann man Texte selbst eintippen, die später nur noch durch die Redaktion ins Magazin eingebaut werden müssen. Das Programm, inklusive Tools und umfangreicher Anleitung, bekommt man für fünf Mark (Vorkasse) bei der Redaktion. "Digital Talk" ist bei PD/Shareware-Händlern erhältlich und ist freikopierbar.

Redaktionsanschrift:
Jörg Nehls, Marienbergstr. 12,
D-31171Nordstemmen
Johannes Rinderer, Eckweg 20,
A-6845 Hohenms

ORIGINAL-SOFTWARE

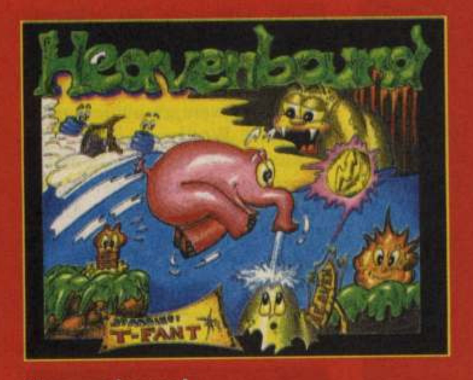
Endlich gibt es sie wieder! Die besten Spiele für den C 64. Direkt

Greifen Sie zu, solange der Vorrat reicht! (Spiele nur noch in begrenzter Menge vorhanden!)



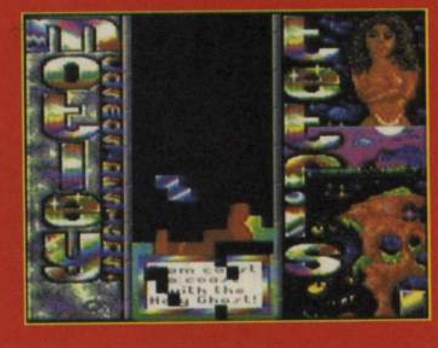
Flight Pack

Mit dem "Flight Pack" haben Sie vier mal die Chance Herr der Lüfte zu sein. Neben "Deep Strike" und "Combat Lynx", auch die knackige Bombersimulation "First Strike".
Best.Nr. 641119
nur DM 9,80



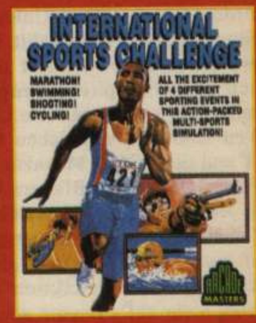
Heavenbound

T-Fant wird gesucht! Begeben Sie sich zu einem heißen Jump'n Run-Abenteuer als Elefanten-Held in Richtung Himmel.
Bestell-Nr. 649410
nur DM 19,80



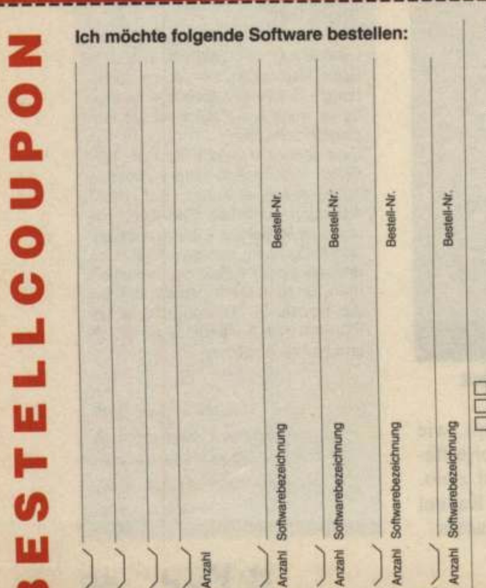
Motley Tetris

Die Variante des Tüftel-Hit aus Russland bunt wie noch nie und garantiert suchterzeugend
Bestell-Nr. 649409
nur DM 9,80



International Sports Challenge

Eine aktionsgeladene Multi-Sport-Simulation für einen oder zwei Spieler: Marathon, Schwimmen, Schießen und Radfahren!
Best.Nr. 649417
Preissensation: DM 9,80



BESTELLCOUPON

Ich möchte folgende Software bestellen:

Nr.	Anzahl	Softwarebezeichnung	Bestell-Nr.
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			

ABSENDER (Bitte leserlich ausfüllen)

Name, Vorname

Straße / Nr.

PLZ Ort

Bitte ausschneiden und absenden an:
ERDEM Development,
Postfach 1823, 84471 Waldkraiburg oder
Tel. 08638 / 9670-70, Fax 08638 / 9670-55

Gewünschte Zahlungsweise bitte ankreuzen:
Ausland nur gegen Vorkasse mit Euroscheck oder Postanweisung; zzgl. DM 12,- (Versand, Porto)

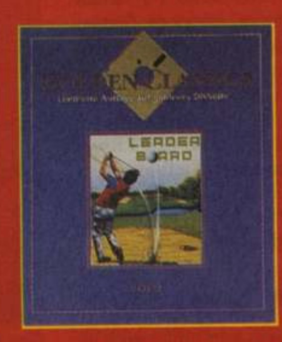
- ☐ Vorkasse mit V-Scheck (Versandkostenpauschale 7,- DM)
- ☐ Per Nachnahme (Versandkostenpauschale 12,- DM)
- ☐ Bankabbuchung (Versandkostenpauschale 7,- DM)

Bankleitzahl

Konto-Nr. Kontoinhaber

Geldinstitut

Datum, Unterschrift (bei Minderjährigen des gesetzl. Vertreters)



Leader-Board Golf

Spielen Sie wie Bernhard Langer oder Nick Faldo auf Ihrem C 64.
Best.Nr. 640702
Preissensation: DM 9,80

ZUM KNÜLLERPREIS!

von uns. So billig war Original-Software noch nie! Packen Sie zu!

TOP-SELLER



Big Box

30 Spiele in einer Box voller Action und Strategie. Darunter viele Klassiker wie Wonderboy, Ghostbuster, Spindizzy, Rampage und Hacker.
Best.Nr. 640601
nur DM 29,90

AUSVERKAUFT



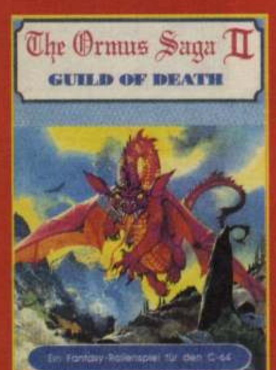
Big Box 2

Phantastische Spiele-Sammlung mit dreißig Titeln: u.a. "Driller", "Hunters Moon", "Gribbly's Day Out" und "Sanxion
Best.Nr. 649401
nur DM 29,90



Sword of Honour

Finden Sie als Ninja das Schwert des Shoguns! Retten Sie seine Ehre und bewahren Sie ihn vor dem Freitot durch Harakiri. Zwei Disketten voll mit Action und vielen Rätseln. Fantastische Grafiken entführen Sie in das ferne Japan.
Best.Nr. 649418
nur DM 29,80



The Ormus Saga II

Kämpfe gegen Drachen, Dämonen, Zauberer und andere Bösewichte und beweise taktisches Geschick in der Schlacht gegen die Truppen des Ormus-Kultes.
Best.Nr. 649404
nur DM 24,90



Soccer Mania

Die Fußballspiel-Sammlung der Extra-Klasse mit "Football Manager 1 und 2 " und dem Ultra-Hit "Microprose Soccer"
Best.Nr. 649402,
nur DM 19,90

NEU!



"Spiele-Koffer"

Drei heiße Games der Spitzenklasse auf einen Schlag: Zak McKraken, Oil Imperium und European Soccer. Der Preisknüller und Geheimtip für Spieler!
Best.Nr. 649411

Preissensation: DM 29,80

NEU!



Ormus Saga III

Der abschließende Teil der Ormus-Saga-Trilogie entführt den Spieler erneut nach Beryland, wo es gilt, zahlreiche Abenteuer zu bestehen. Auch für Neueinsteiger interessant, da man keinen der beiden Vorgänger kennen muß!
Best.-Nr. 649407
Preis: 39,90 Mark

NEU!



Magic of Endoria

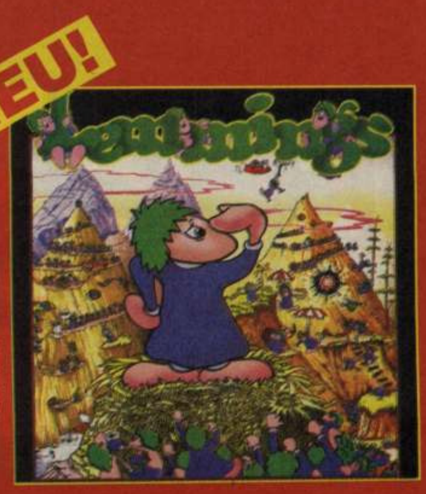
Graben Sie im Erdreich von Endoria nach den geheimnisvollen Kristallsplittern und steigen Sie zum Gott auf. Eine Mischung aus Fantasy und Handelssimulation.
Best.-Nr. 649406
Preis: 49,90 Mark

NEU!

Lemmings

Die dümmlichen Nager sind da und verbreiten auf fünf Disketten Spielefieber und Suchterscheinungen. Bringen sie die kleinen Kerle mit der richtigen Strategie heil durch die 120 Level.
Best.Nr. 649405

Preissensation: DM 29,80



Szene Inside

Weil die 64'er-Szene scheinbar noch nicht aus dem Sommerurlaub zurück ist, gibt's diesmal keine 64'er-Charts. Dafür einige heiße Infos zu neuen Produktionen von Plush, Aspheron und Agony.

Das brandneue Demo "Love" von Agony geizt auf zwei Disketten nicht mit Grafiken, Effekten und tollen Sounds. Vektorräume, Voxel-Space-Landschaften, Bitmap-Effekte und höllisch schnelle Riesenscroller sind weitere Highlights von "Love". Die Grafiken und Fantasy-Bilder kommen von den Agony-Mitgliedern Astaroth und Roder.

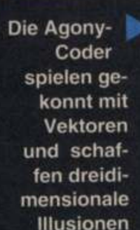
Dem stehen die Jungs von Plush mit ihrer Sound-Sammlung "Plush Ethics" kaum hinterher. Schöne IFLI-Logos, integrierte IRQ-Loader mit Animationen und ein Auswahlmenü inklusive "virtueller" Musikbox machen das Hörvergnügen bei "Plush Ethics"perfekt. Ein neues Demo mit dem Titel "PLUSH-WORLD" haben die Plush-Mitglieder schon in Vorbereitung.



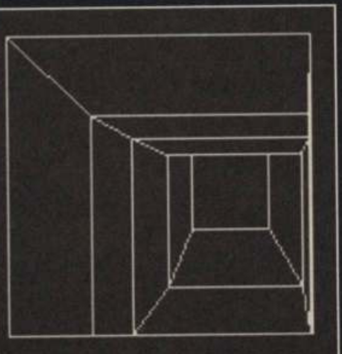
Egal ob Single- oder Multicolor-Modus - die Agony-Logos sind einfach sehenswert



Drachen sind scheinbar die Lieblingstiere der Agony-Freaks



Die Agony-Coder spielen gekonnt mit Vektoren und schaffen dreidimensionale Illusionen



Astaroth-Grafik: Wenn der Schwarze Ritter lacht, werden alle umgebracht!





Der Drachen fordert zum Wenden der Diskette auf - das Demo "Love" nimmt zwei Floppy-Seiten ein



Fantasy in FLI: ein Kunstwerk des Agony-Grafikers Roder





Ganz frisch ist das Demo "LILITH", der vor kurzem gegründeten Gruppe Aspheron, deren Mitglieder aus Deutschland und Polen kommen. Stretch-Effekte und Vektor-Routinen werden bei den Aspheron-Leuten ganz groß geschrieben.

"Plush Ethics" ist eine digitale Musikbox mit tollen Grafiken, Animationen und natürlich vielen Sounds

Minis





VORSCHAU  10/95

SIE KOMMT ZU IHNEN INS HAUS AM 29.9.95

Schwerpunkt: Anwendungen

Packet-Radio und CB-Funk

Schaufeln Sie mit Ihrem C 64 Daten über den Äther! Wir zeigen Ihnen, wie es funktioniert.

C 64 und Taschenrechner

Datentransport zwischen Pocket-Computer und dem C 64 leicht gemacht – Grundlagen und Bauanleitungen im Heft.

CD-Commander

Wir testen die C-128-Software zum Lesen von CD-ROMs!



Text-Editor auf Disk

***WINScript:* eine neue leistungsfähige Textverarbeitung mit vielen Funktionen und komfortabler Benutzeroberfläche und integriertem Zeichensatz-Editor auf Diskette im Heft.**

```
Zeile: 1      Seite: 1

_IN - Script  V 2.1

WIN-Script ist aus seinen 'Kinderschuh
neuesten Version 2.1 vor. Es handelt s
sondern um eine völlig überarbeitete V
anderen Texteditoren scheut, ja wohl z
C64 gehört.

WIN-Script bietet neben umfangreichen
ten Zeichensatz- Editor sowie eine her
sind die Ganz- Seitendarstellung, das
und die Übernahme und Anpassung fremde

WIN-Script ist durch die ausschließlic
äußerst schnell und leistungsfähig.
```

Datentransfer: Files von fremden Systemen

Wir führen Sie ein ins "Im- und Export-Geschäft" für Computer-Dateien. So nutzen Sie alle Möglichkeiten, Programme und Files von anderen Computer-Systemen zu übertragen, um daraus eine waschechte C-64/C-128-Datei zu machen! Die entsprechende Software finden Sie wie gewohnt auf der Programmservice-Diskette zum Heft.

Inserentenverzeichnis

Unsere heutige Ausgabe enthält Beilagen der Firma Stonysoft, Babenhausen.

PPE HARDWARE ORDERS: 02227 3221 SOFTWARE PPE

64NET-SERIE

64NET, Vollversion	**109,00**
64NET Demoversion	5,00
64NET Verb.-Kabel (1,2m)	39,90
5/10/15/30m	59/89/129/189,00

Bitte PC-Diskettenformat angeben

CD-COMMANDER

CD Commander 128 V1.x	**39,00**
CD Connector/Kabel	39,90
CD-Komplettpaket	429,00

=SCSI CD-ROM-Lfw.,ext. Geh.,NT/Trafo,Kabel,Software

CD-Audioplayer	xx,xx

RAM-ERWEITERUNGEN

RAMDrive 2Meg	499,00
BBGRam 1 Meg	279,00
BBGRam 2 Meg	369,00
REU 1764 (incl. MAVERICK "lite")	99,00

CBM-FLOPPY-LFW.

Floppies ohne NT aber mit MAVERICK "Lite"-Version

1581 3,5" Lfw.	**300,00**
1571 5,25" Lfw.	280,00
1541-II 5,25" Lfw.	210,00

PC64-KONFIGURATIONEN

Minimal-Konfiguration (C64, ohne Floppy, mit Speeder, NT, Tastatur)	**499,00**
Standard-Konfiguration (C64, Floppy HD1581, Speeder, NT, Tastatur)	649,00
Advanced-Konfiguration (C64, Floppy HD1581 & 1541, Speeder, NT)	799,00
Proffessionell-Konfig. (C64, Floppy HD1581 & 1541, Speeder, NT, Sound, RTC)	899,00

POWERSUPPLIES

Netzteil f. C64	39,00
Netzteil f. RAMDrive	15,00
Netzteil f. BBGRam	10,00
Netzteil f. BBU	10,00
Netzteil f. CBM 1541/71/81	49,00

TELEKOMMUNIKATION

Modem CBM 1200 bps	39,90
Modem ALTEC 14.4 bps	219,00
DATABlast ModemLink	**89,00**
RS 232 Schnittstelle	49,00
Schnittstellenkabel	a.A.
DREWS Dekoder 1.6_up	9,90
DREWS Dekoder 1.6	19,90
BTX-EXTRA 64	**19,90**
BTX-EXTRA 128	**29,90**
NOVATERM 9.5_GE	69,00
KOM-Pack I-1200 (=1200 Modem CBM1670,BTX-Sw,DFÜ-Sw)	99,00
KOM-Pack II-14.4 (=14.4ALTEC Modem,DATABlast-Modul,BTX-Sw,DFÜ-Sw)	329,00

GIG GEOS-HEFTE

Geopublish SH 1	19,80
Geopublish SH 2	19,80
Geopublish SH 5/NEU!/	**19,80**
Geofile SH	19,80
Geocalc SH	19,80

PC GEOS

PCGEOS Ensemble 2.01/US	**169,00**
PC GEOS 2.01 dtsch. Vers.	399,00
PC GEOS Geopublish Demovers.	10,00

zum Kennenlernen:
PC GEOS Geopubl. & das große PC GEOS Buch nur 39,00

GEOS DIV.

GEOS GAMES Vol 1 - 4 je	**19,00**
GEOSHELL 2.2_GE	**50,00**
GeoCanvas 3.0	59,00
GeoCOM 1.5	59,00
GEOS LQ 2.1 ab	49,00
TextPrint V3	34,00
GeoCOM Sampler 3,5"Disk	**75,00**
The Best of Collection 3,5"Disk	45,00
FIRST AID -Erste Hilfe f. GEOS	29,00
geoLabel	22,00
geoPrint	**19,00**
Landmark Series	45,00
TopDesk 64/128	21,00
geoMakeBoot	23,00
gateWay 64/128	60,00
PP Grafik Collection Abo f. 1 J.	99,00
GEOS Compagnion	49,00

SOFTWARE BASIC DIV.

Multi-Lager	29,00
Multi-Datei-System	29,00
Kombitext (Textverarb.)	19,00
Ghostwriter System 3	49,00
MASTER 64	19,80
GoDot/Vollversion/NEU!	**49,00**
'81 TOOLKIT	49,00
MAVERICK V4	**50,00**
CD-ROM Brotkasten Live Vol.1	**49,00**
MATTINGs PD-BOX Software	**19,00**
Eddison Zeichenprg.	58,00

HARDWARE DIV.

BBRTC	49,00
Maus f. C64/128/GEOS	60,00
Maus/Uhrset	94,00
VDC-Upgrade-KIT	79,00
BBU	**99,00**
AR MKVI-Modul	119,00
AUTO 40/80 Videoswitch	109,00
Joystick/Joypad ab	**19,00**
KeyDOS V2	**60,00**
FinalChessCard/Schach f. C64	39,00
geoCABLE	29,00
Userportswitch 2fach	99,00
Flash 8/256K/1000K	**349/449,00**
WIESEMANN Druckerinterface	69,00
EX-3 Expansionsportweiche	**65,00**
HERKULES neue Monitorlösung	**98,00**
LayoutDesigner Modul	78,00
TurboDOS Floppyspeeder	99,00
MasterModul Ass. Exbasic, Zubeh.	89,00

SCANNER, DTP, VIDEO

Handyscanner 64 incl.GEOSSW.	298,00
Superscanner III	298,00
PAGEFOX	228,00
Genbox	498,00

GEOS STANDARD

GEOS 64 2.5	89,00
GEOS 64 2.5 Update 2.0/1.x	49/69,00
Geocalc 64	59,00
Geofile 64	59,00
GEOS 128 2.0	119,00
Geofile 128	79,00
Geocalc 128	79,00
Geochart 64/128	49,00
Geopublish 64/128	59,00

nicht vergessen: **GAMES-CORNER**

unsere Hausmarken...





RAMDrive
Speicherriese incl. JiffyDOS Universell für GEOS u. BASIC! Akku/Netzteilgepufferte RAM-Erweiterung mit Swapfunktion, Resetbutton, EIN/AUS-Switch, incl. GEOS/Basic-Software

BBGRam
ideale Speichererweiterung für GEOS 64/128, batterie-/netzteilgepuffert, GEOS-Start aus der RAM heraus! Daten bleiben auch nach dem Ausschalten des C64/128 erhalten!





BBU
RAM-Mem-Saver-Cartridge für C64/128 RAM-Erweiterungen; batterie/Netzteilgepuffert; speichert RAM-Daten auch noch nach dem Ausschalten des Computers - ein MUß!

PC64-Mini-Tower
*200 Watt Netzteil
*integrierte Disk-Lfw. (optional)
*neues Betriebssyst./Speeder
*externe Tastatur
*modifizierter Userport
*Reset-Taster



mehr als 2000 64'er Programme auf einer CD incl. Emulatoren



...das Ei des Kolumbus?
Der CD-Commader erlaubt den Zugriff auf an CMD-HD's angeschlossenen SCSI CD-ROM-Lfw.; diese Software kopiert Daten einer CD (z.B. 64'er CD-ROM) auf CBM/PPE/CMD-Lfw.! Hardwarevoraussetzungen (nur C128/80Z): CMD-HD-Festplatte, SCSI CD-ROM-Lfw., Netzteil, Verb.-Kabel. *CD-ROM-Lfw. an CMD-Festplatten, jetzt mit CD-Commander*
CD-Komplettpaket f. C128 nur 429,00 DM - Info anfordern

64NET Client/Server-Software für Commodore 64 &128

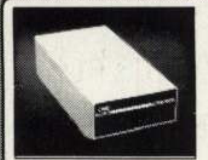
64NET - Nutzt die Festplatte eines jeden PC/AT 286 und besser um dort Daten zu speichern oder zu lesen. Ideal für GEOS aber auch für BASIC 2.0/7.0 und goDOT. Verbindung zum PC über Userport. **Datenübertragung bis zu 40 Kb/Sek.!** Bis zu 8x schneller! als eine CMD-Festplatte! Richten Sie beliebig viele CBM-Disk-Emulationen unter GEOS ein und verwalten diese komfortabel! 100% kompatibel zur CBM-Software.
Die Revolution in Sachen Massenspeicher für den C64/128.

PPE PERFORMANCE PERIPHERALS EUROPE
Silcherstraße 16
D-53332 Bornheim

Fon & Fax: 02227 3221
BTX: *MATTING#
Produktliste gratis

Bestellung, Versand, Geschäftsbedingungen
per Nachnahme zzg. + 10,00 per Vorkasse zzg. + 6,50
Wir akzeptieren: bar, EC zur Verrechnung, VISA, EURO, AMEX
Preise in DM, Angebot freibleibend, Änderungen vorbehalten.
Herstellerbedingte Lieferzeiten möglich.

CMD - Das Höchste in Sache C64/128-Leistung



HD Series
HARD DRIVES

Das letzte Wort im Groß-Speicher für den C-64/128

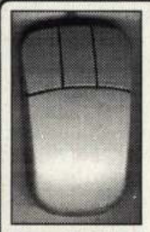
Hier haben Sie Großspeicherfestplatten für Ihren C-64 und C-128. Sofort in Betrieb zu nehmen mit leichter Installation via Seriellport. Die HDs werden mit einer Echtzeituhr, Dienstprogrammen, Kopierprogrammen, und deutschem Handbuch geliefert. Voll kompatibel mit GEOS, CP/M, BBS und vielen anderen Applikationen. HD Series-Festplatten sind in 40 MB, 170 MB, 340 MB, 500 MB, 1 GB Kapazitäten erhältlich.





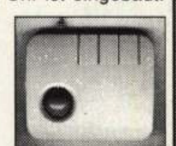
Großkapazitätslaufwerke für 1,6- und 3,2- MB-Disketten

Die leistungsstarken neuen 3,5" Laufwerke von CMD sind rückwärtskompatibel mit Commodore's 1581 (800K)Laufwerk. Diese Laufwerke tun aber viel mehr! Beide Modelle haben einen 1,6MB Speichermodus und die FD-4000 bietet zusätzlich einen 3,2MB-Modus an. Beide Geräte sind JiffyDOS kompatibel (LaufwerksROM eingebaut) und haben eine Echtzeituhroption. Die FD-Series ist perfekt für GEOS, BBS-Betrieb und vieles mehr.



CMD - SMART Maus und SMART Trackball

Kein 1351-Klone! Es handelt sich hier um proportionale Eingabegeräte, die viel mehr können! Die Doppelklick-Funktion ist selbstverständlich dabei und eine Echtzeituhr für automatisches Setzen der GEOS-Uhr ist eingebaut. Beide Geräte sind mit der 1351 voll kompatibel und daher für andere Programme geeignet. Akkugepuffert und mit einer Utilities-Diskette und deutscher Anleitung geliefert. Sie sind zweifellos die Eingabegeräte der Gegenwart und Zukunft.





RAMLink™

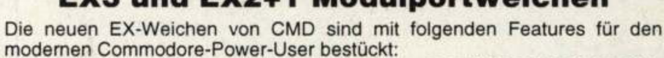
Akkubetriebenes und erweiterbares RAM-Laufwerk

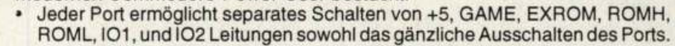
RAMLink gibt Ihrem Computer ein schnelles RAM-basiertes Laufwerk, das Programme und Daten auch nach Ausschalten des Computers speichert. Der als Option angebotene Akku bietet Ihren Daten bei Stromausfall Sicherheit. Die RAMLink läßt sich auf 16MB erweitern, hat einen Pass-Thru-Port für Modulgebrauch und einen RAM-Port für den Gebrauch mit vorhandenen RAM-Erweiterungen (REU oder GEORAM). Ein Parallelanschluß ermöglicht Hochgeschwindigkeitsdatenübertragungen von und zu CMD's HD. Eingebaute JiffyDOS-Rountinen erleichtern Diskettenzugriffe und bescheunigen den Laufwerksbetrieb mit Systemen, die mit JiffyDOS ausgestattet sind. Einmalig in der Zusammenarbeit mit GEOS, Textverarbeitungen, Telekommunikationssoftware und mehr.



CMD Direkt

RAMLink	
RAMLink 1 MB mit Echtzeituhr	DM 465,-
RAMLink 4 MB mit Echtzeituhr	DM 699,-
RAMLink Puffer-Batterie	DM 70,-
Parallelkabel	DM 35,-
1MB/4MB SIMM	DM 89/299,-
HD-Serie Festplatten	
HD-40, 42 MB SCSI-Festplatte	DM 660,-
HD-80, 80 MB SCSI-Festplatte	DM 719,-
HD-100, 170 MB SCSI-Festplatte	DM 799,-
HD-200, 270 MB SCSI-Festplatte	DM 899,-
HD-500, 500 MB SCSI-Festplatte	DM 999,-
HD-1000, 1 GB SCSI-Festplatte	DM 1499,-
FD-Serie Laufwerke	
FD-2000 Großspeicherlaufwerk (1,6MB)	DM 399,-
FD-2000 mit Echtzeituhr	DM 425,-
FD-4000 Großspeicherlaufwerk (3,2MB)	DM 529,-
FD-4000 mit Echtzeituhr	DM 555,-
10er-Packung, HD-Disk. (1,6 MB)	DM 35,-
10er-Packung, ED-Disk. (3,2 MB)	DM 85,-
JiffyDOS* (Bitte Pinzahl angeben C64)	
C64-System (24 oder 28 Pin Kernal)	DM 75,-
SX-64-System	DM 75,-
C-128-System	DM 99,-
128-D-System (Metallgehäuse)	DM 99,-
128-D-Tragbar (Kunststoffgehäuse)	DM 99,-
Zusätzliche Laufwerke-ROM's	DM 40,-
Weitere Soft- u. Hardware	
CMD-SmartMaus	DM 99,-
CMD-SmartTrackball	DM129,-
GEOS Paralleldruckerkabel	DM 59,-
EX3/EX2+1 Modulportweiche	DM 69/79
CMD GAMEPAD	DM 49,-
SwiftLink-RS232	DM 99,-
SID Symphony Stereo Cartridge	DM 99,-
Big Blue Reader	DM 55,-
Compression Kit'94	DM 75,-
geoMakeBoot	DM 22,-
gateWay/64 oder gateWay/128	DM 60,-
geoProgrammer (ML für GEOS)	DM 99,-
GEOBasic (BASIC für GEOS)	DM 45,-
Collette Utilities	DM 60,-
GEOS Power Pak I o. II/Companion je	DM 45,-
IPAINT	DM 79,-
IPORT	DM 65,-
GEOS & GEOS Applications (US)	Anrufen





NEUE EX2+1

Die neuen EX-Weichen von CMD sind mit folgenden Features für den modernen Commodore-Power-User bestückt:

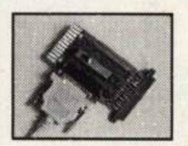
- Jeder Port ermöglicht separates Schalten von +5, GAME, EXROM, ROMH, ROML, IO1, und IO2 Leitungen sowohl das gänzliche Ausschalten des Ports.
- Der Mittelport besitzt einen IO1/IO2 ($DE00/$DF00) Swapschalter.
- Reset-Schalter

Die EX3 ist für den Gebrauch mit SwiftLink, SID Symphony, REUs, GEORAM und Utility-Modulen geeignet. Die EX2+1 ist der EX3 identisch, hat aber 2 vertikale und einen horizontalen Port.

SwiftLink-RS232 Modul

SwiftLink bietet dem C64/C128 das an, was andere Computer seit Jahren besitzen: einen echten RS-232 Port. SwiftLink unterstützt Datenübertragungsgeschwindigkeiten von bis zu 38,400 BAUD und ermöglicht das Anschließen moderner Hochgeschwindigkeitsmodems an den Commodore. SwiftLink kann auch als Nullmodem mit anderen Computern für schnelle Datenübertragungen benützt werden. SwiftLink wird mit DFÜ-Software und Dateitransfersoftware geliefert, um das Konvertieren von PET ASCII auf Standard-ASCII zu ermöglichen. Voll kompatibel zum neuen GEOFAX-Programm.





GEOCABLE II Paralleldruckerkabel

Das fortgeschrittenste Paralleldruckerkabel, was je gebaut wurde! Diese durchgeschliffene Userport-Weiche besitzt einen Pass-Thru-Switch, um zwischen Drucker und einem anderen angeschlossenen Gerät ohne Konflikt zu schalten. GEOS-Treiber und Utilities werden mitgeliefert. Voll kompatibel mit u.a. GEOS, Superbase, SuperScript, GEOS-LQ, Action-Replay MK VI.

SID Symphony Stereo Modul

SID Symphony verdoppelt die Soundstimmen Ihres C64ers/C128ers. SID Symphony kann mit dem Ausgang Ihres Computers kombiniert oder an Ihr Stereogerät angeschlossen werden. Der SID Stereo Player und Demo-Songs werden mitgeliefert.



CMD GAME - PAD



Haben Sie es mit den traditionellen Joysticks satt? CMDs neuer SEGA-ähnlicher Game Pad bietet bequeme Feuerkontrolle mit Turbofunktion, einen Daumen-Kontrollpad, der sich in 8 Richtungen bewegen läßt, und ein 3 Meter langes Kabel. Unsere Kunden schwärmen von diesem neuen Produkt!

* Es ist wichtig, Ihre Computer- und Laufwerksmodellnummer bei der Bestellung anzugeben. C64-Besitzer müssen die Seriennummer auch angeben. Ein JiffyDOS-System beinhaltet die ICs für einen Computer und ein Diskettenlaufwerk.

Preise enthalten Fracht, Zoll und Steuer. Lieferung ca. 4 bis 6 Wochen. Preise können ohne Vorankündigung geändert werden. Bei Vorkasse keine Versandkosten. NN + 10,00 DM

CMD Direkt

CMD Direkt Sales
Postfach 58
A-6410 Telfs, Austria
Tel/FAX: 0043-5262-66080
BTX: *MATTING#